18 OCTOBER 1979

AUTOMATION TECHNOLOGY
(FOUO 3/79)

1 OF 1

JPRS L/8720

18 October 1979

USSR Report

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 3/79)

FBIS FOREIGN BROADCAST INFORMATION SERVICE

NOTE

JPRS publications contain information primarily from foreign newspapers, periodicals and books, but also from news agency transmissions and broadcasts. Materials from foreign-language sources are translated; those from English-language sources are transcribed or reprinted, with the original phrasing and other characteristics retained.

Headlines, editorial reports, and material enclosed in brackets [] are supplied by JPRS. Processing indicators such as [Text] or [Excerpt] in the first line of each item, or following the last line of a brief, indicate how the original information was processed. Where no processing indicator is given, the information was summarized or extracted.

Unfamiliar names rendered phonetically or transliterated are enclosed in parentheses. Words or names preceded by a question mark and enclosed in parentheses were not clear in the original but have been supplied as appropriate in context. Other unattributed parenthetical notes within the body of an item originate with the source. Times within items are as given by source.

The contents of this publication in no way represent the policies, views or attitudes of the U.S. Government.

For further information on report content call (703) 351-2938 (economic); 3468 (political, sociological, military); 2726 (life sciences); 2725 (physical sciences).

JPRS L/8720

18 October 1979

USSR REPORT

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 3/79)

This serial publication contains articles, abstracts of articles and news items from USSR scientific and technical journals on the specific subjects reflected in the table of contents.

Photoduplications of foreign-language sources may be obtained from the Photoduplication Service, Library of Congress, Washington, D. C. 20540. Requests should provide adequate identification both as to the source and the individual article(s) desired.

CONTENTS PAGE

UDC 681.32(0.75)

CONTROL MACHINES AND THEIR UTILIZATION

Moscow UPRAVLYAYUSHCHIYE MASHINY I IKH PRIMENENIYE in Russian 1978 pp 179-231

[Table of Contents and Chapters 5 and 6 from the book by R.P. Stroganov, Izdatel'stvo Vysshaya Shkola]

[Text] TABLE OF CONTENTS

CHAPTER 5. WAYS OF INCREASING THE RELIABILITY OF UVM'S

5.1. Reliability: Basic Concepts and Definitions; Ways of Guaranteeing Reliability

Reliability is the capability of a system (element) of carrying out its assigned functions under certain operating conditions. No technical device possesses absolute reliability, so failures and malfunctions are possible when it is in operation. A failure is a disruption of normal operation during which the system (element) completely or partially loses its ability to perform its assigned functions. A malfunction is a self-eliminating failure caused by the effect of different types of interference and internal noise on the system.

The different types of reliability are: equipment, functional and operational. The equipment reliability of a UVM characterizes its ability to carry out its assigned functions and takes equipment failures into consideration. The functional reliability of a UVM is determined by the probability of the UVM's correct performance of its functions under conditions of failures and malfunctions. This reliability is determined not only by the degree of perfection of the UVM's equipment realization, but also by the characteristics of the monitoring systems that are used. The operational reliability of a UVM is determined both by the degree of perfection of the equipment and the monitoring system and the suitability for repair of the device in question.

It is possible to distinguish the following ways of increasing the reliability of UVM's (or any technical device): 1) improve the quality of the manufacturing process and the principles of the construction of the system's elements; 2) improve the structure and introduce redundancy; 3) improve reliability during the operating process.

The first two methods of improving reliability are used during the process of planning and manufacturing both the elements and the system as a whole. As a result, the system achieves a certain level of reliability. The third method basically consists of insuring the level of reliability achieved as the result of the development and production of the technical system.

The basis for insuring the reliability of any complex system is the use of reliable elements to manufacture the system. The elements' reliability is arbitrarily evaluated by the intensity of failures λ_{el}.

Let us discuss the requirements for the intensity of element failure that are made on modern, complicated information processing systems, in which category UVM's are included.

If we take the total number of elements in such systems to be $n_{el} = 1 \cdot 10^6$ and require that the average time of failureless operation T_0 be at least $1 \cdot 10^4 - 1 \cdot 10^5$ hours (1-10 years), an approximate estimate of the intensity of element failure can be obtained in the following manner.

The average time of failureless operation is

$$T_0 = \int_0^\infty t \frac{dq(t)}{dt} dt = \int_0^\infty t \frac{d(1-P(t))}{dt} dt = \int_0^\infty t dP(t), \qquad (5\text{-}1)$$

where q(t) = probability of failure.

Integrating (5-1) by parts, we obtain

$$T_0 = -tP(t)\Big|_0^\infty + \int_0^\infty P(t)\,dt. \qquad (5\text{-}2)$$

Let us assume that over the course of time T_0, λ_{el} = constant. In this case, the probability of failureless operation is governed by an exponential law; that is,

$$P(t) = e^{-\lambda_{el}t}.$$

For such a law, $\lim e^{-\lambda_{el}t} = 0$, since the rate of decrease of P(t) is greater than the rate of increase of t. Considering what has been said, (5-2) can be written in the form

$$T_0 = \int_0^\infty P(t)\,dt.$$

Systems with λ_{el} = constant, for which the operational failure of one element leads to the operational failure of the entire system, are characterized by

$$T_0 = \int_0^\infty e^{-n_{el}\lambda_{el}t} dt.$$

For $n_{el} = 1\cdot10^6$ and $T_{0\ sup} \geqslant 1\cdot10^4 - 1\cdot10^5$ hours,

$\lambda_{d} \leqslant 1\cdot10^{-10} \div 1\cdot10^{-11}$ failures/hr.

The technology and principles of element construction existing at the present time make it possible to obtain indices of reliability with respect to failure intensity that are one or two orders less than this value. In some cases, therefore, it is impossible to satisfy the requirements for operational reliability of a system without using structural methods to increase it.

The structural methods provide for: 1) the use of optimally reliable structures (those that insure performance of the given functions with minimal system complexity); 2) the introduction of redundancy into the system's structure in cases where the optimum structure does not guarantee the required reliability index value. In control systems that contain UVM's and are used for general industrial purposes, it is possible to raise the reliability level basically because of the introduction of redundancy into the system's structure through redundancy of individual units or the machine as a whole.

Along with the structural methods, it is possible to increase reliability by improving the reliability characteristics of a UVM through the use of improved monitoring methods, improved maintenance, and the use of automatic methods for searching for malfunctions.

The developers of control systems based on existing UVM's have at their disposal only structural and operational methods for increasing reliability, since the introduction of equipment changes in a computer's design on an operational basis is, as a rule, extremely undesirable. The methods of redundancy and program monitoring are the ones most widely used in such systems.

5.2. Determining the Required Degree of Reliability of a UVM

Before evaluating the possibilities of using different redundancy and program monitoring methods to increase the reliability of UVM operation, let us discuss the required level of reliability of a UVM as an element of a specific control system.

The level of UVM reliability cannot be chosen arbitrarily. It depends on the level of reliability of the other parts of the system. As the requirements for UVM reliability become more rigorous, its cost increases, and at the same time it may prove to be the case that the resulting reliability of the system as a whole is limited by either the object of control, or the sensors, or the communication channels.

The probability of system failure for the case where the failure of one of its parts results in the failure of the system as a whole is

$$q^c(t) = 1 - \prod_{i=1}^{n} P_i(t), \qquad (5\text{-}3)$$

where $P_i(t)$ = probability of failureless operation of the i-th part of the system over some period of time.

If we separate the probability of failureless operation P_2 of the control unit, then for some fixed t it is possible to write (5-3) as

$$q^c = 1 - P_1P_2,$$

where P_1 = probability of failureless operation of the object.

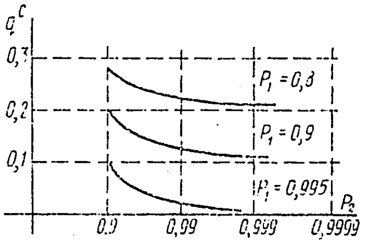


Figure 5-1. Dependence of probability of failure of a system on the operational reliability of the computer and the controlled object.

The dependence of q^c on P_1 and P_2 usually has the form shown in Figure 5.1.

With the exception of cases where the operational failure of the control unit can entail injuries to humans, requirements for UVM reliability should not exceed a certain level. Since an increase in a system's reliability is inevitably related to an increase in its cost, techniques for determining the necessary level of UVM reliability should be based on optimization of the system relative to the criterion of minimum expenditures for the given limitations.

The complete expenses C for a system can be defined in the form

$$C = C_{cp} + C_o + C_{ls}.$$

where C_{cp}, C_o = capital and operating expenses, respectively; C_{ls} = losses arising as the result of unreliability of the system.

Capital expenses contain two components: 1) C_{cp1}, which does not depend on reliability, and 2) C_{cp2}, which is a function of reliability; that is,

$$C_{cp} = C_{cp1} + C_{cp2}.$$

The C_{cp2} component can be evaluated by the formula [10]

$$C_{cp2} = C_{cp3}(q_{kn}/q)^{2}. \qquad (5\text{-}4)$$

where C_{cp3} = the component of capital expenses that depend on the reliability in the existing (known) system (computer); q_{kn}, q = probability of failure in the existing (known) system and the one being developed, respectively; α = a coefficient that is determined experimentally.

From (5-4) it follows that if $q \to 0$, then the additional capital expenses $C_{cp2} \to \infty$; that is, they increase extraordinarily rapidly as the requirements for the system's operational reliability are raised.

The operating expenses can be defined in the form

$$C_o = (C_{cp} + C_{mp} + C_{mt})\, t,$$

where C_{en}, C_{mp}, C_{mt} = costs of the energy, manpower and materials, respectively, expended to operate the system per unit of time; t = time of operation of the system.

Losses from system failure C_{ls} over time t are determined by the average cost of the losses incurred for one failure C_{av} and the mathematical expectation of the number of failures $M[N_0]$; that is,

$$C_{ls} = c_{av} M[N_0], \tag{5-5}$$

it being the case that $c_{av} = \frac{1}{M[N_0]} \sum_{i=1}^{m} c_i$, where c_i = cost of losses for the i-th failure of the system (machine).

In the first approximation, a stream of failures in a system can be regarded as the simplest and distributed according to Poisson's law. For such a stream, the probability of the occurrence of m failures in time t is

$$q(t) = \frac{a^m}{m!} e^{-a},$$

where $a = \lambda^c t$ and λ^c = intensity of system failures.

If the probability of failure is low, which is equivalent to a low value for a, then

$$q(t) = \lim_{m \to m_{min}} \frac{a^m}{m!} e^{-a} \approx a = \lambda^c t.$$

For Poisson's law, the mathematical expectation of the number of failures is $M[N_0] = a = \lambda^c t$, so (5-5) can be written as

$$C_{ls} = c_{av}\, q(t).$$

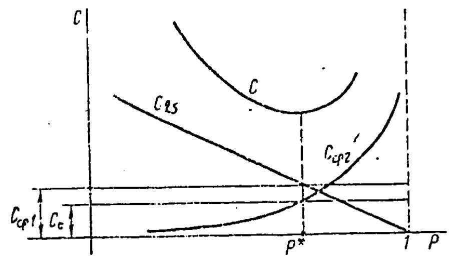


Figure 5-2. Toward determining the optimum level of UVM reliability in a control system.

For a given value of t, $q(^0t) = q = \lambda^c t$ and $C_{ls} \approx C_{av}q$. Thus, the total expenses for the system are

$$C = C_{cp1} + C_{cp3}(q_{Kn}/q)^{\alpha} + C_c + c_{Av}q. \quad (5\text{-}6)$$

The dependence of expenses C on the level of reliability $P = 1 - q$ is shown in Figure 5-2.

The system's optimum level of reliability can be determined by solving the equation $dC/dq = 0$ for q. As a result of the solution, we obtain

$$q^* = (\alpha C_{cp3} q_{kn}/c_{Av})^{1/(\alpha+1)}$$

or

$$P^* = 1 - q^*,$$

where P* = optimum value of the probability of failureless operation of the system.

The optimum level of system reliability can also be determined on the basis of the conditions for producing the maximum gain. The specific values of P* will, naturally, depend on the criterion that is used.

Knowing the optimum level of system reliability, as well as the reliability levels of its component parts (in addition to the UVM), it is also possible to determine the required optimum level of UVM reliability:

$$P^*_{UVM} = P^*/\prod_{i \in UVM} P_i.$$

5.3. Procedures for Redundancy in UVM's and Operating Modes of Redundant Systems

Redundancy is based on the principle of using excess elements that are not functionally required for normal operation but are intended only to replace the basic elements in case they fail.

Depending on the way the reserve elements are connected, we distinguish total, separate and mixed redundancy. Total redundancy means redundancy of the entire system as a whole. Separate redundancy consists of redundancy of individual elements of the system. Mixed redundancy means redundancy of both individual elements and of the entire system.

Thanks to its simplicity, total redundancy is the variety that is used most widely. In control systems with UVM's, total redundancy is most frequently realized by duplication of the basic computer with an analogous reserve unit.

Depending on the way the reserve element is hooked up, redundancy can be either permanent or by replacement.

For permanent redundancy, the reserve computer is connected to the basic one whenever the latter is operating and is in the same operating mode. Permanent redundancy is used in cases where even a brief interruption in a system's operation cannot be allowed. With this redundancy method, failure of one of the computers should not affect the system's capability to continue performing its functional assignments. The advantages of permanent redundancy are the simplicity of its organization and the lack of work interruptions, which are inevitable when computers are switched; the disadvantages are the increased drain on the reserve computer's service life and the necessity of insuring the system's ability to function when the permanently connected reserve computer suffers operational failure.

For redundancy by replacement, the computer that fails is first disconnected and an analogous reserve computer is then connected. The computer that failed can be disconnected and the new one connected either manually or automatically. The basic advantages of redundancy by replacement are preservation of the reserve equipment's service life, since it can be kept under a light load or none at all; the possibility of using a single computer to back up the control units of several monotypical systems with UVM's, since the probability of simultaneous operational failure of several computers is very low; the lack of any need to correct the operating mode and the parameters of the input and output circuits to which the computer is connected. The disadvantages are the necessity of including switching units with low probabilities of failure in the design of the system; the organization of a developed system for monitoring and indicating irregularities; redundancy not only of the equipment, but also of the information without which the reserve UVM cannot operate when it is connected to the system.

The presence of additional switching devices with limited reliability lowers the overall reliability of the standby system slightly, while the need for redundancy of both equipment and information makes the algorithm for interaction of the operating and reserve computers considerably more complicated. Possible variants for the organization of computer interaction when redundancy by replacement is utilized will be discussed below.

Redundancy by replacement is used to back up systems with UVM's only when some interruption in the operation of the control unit is allowed; this interruption is needed in order to switch the computers from the operating mode into the reserve mode or to search for and eliminate defects, and vice versa.

Regardless of the type of redundancy, its organization requires the inclusion in the system of additional equipment that insures the functioning of the computer complex formed as the result of the redundancy. The amount of additional equipment depends on the redundancy method used. It is most significant for redundancy by replacement. As a rule, this equipment is not mass-produced, but requires special development.

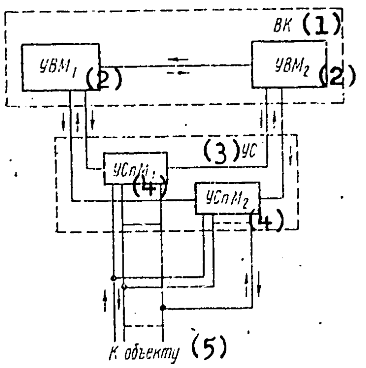


Figure 5-3. Block diagram of a system consisting of two computers connected in the permanent redundancy mode.
Key:
1. Computer complex VK
2. UVM
3. Communication unit US
4. Computer interface device USpM
5. To controlled object

Permanent redundancy, organized in the form of total redundancy by duplication of the computers, assumes complete "equality of rights" of the operating computers. For this redundancy method, there is no separation of computers into basic and reserve categories. Both computers realize identical functional algorithms. As a rule, all machine assemblies -- processor, auxiliary storage, communication links with the controlled object -- are duplicated. Units for monitoring the results of the computation process are used as additional devices that are included in such a system. Figure 5-3 is a block diagram of a system consisting of two computers operating in the permanent redundancy mode.

The system consists of computer complex VK and communication unit US. The US contains computer interface devices USpM$_1$ and USpM$_2$, with the help of which the UVM's input and output circuits are linked.

In each of the system's computers, the same problem is solved independently, using the same data. In order to eliminate the effect of external noise, each of the computers usually utilizes an asynchronous operating mode. The results of the solution are compared, for which purpose the interface devices USpM$_1$ and

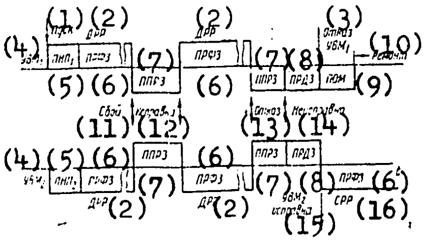


Figure 5-4. Temporal diagram of the operation of a redundant system using the permanent redundancy method.

Key:

1. Start
2. Duplex operating mode DRR
3. Failure of UVM_1
4. UVM
5. Initial start subprogram PNP
6. Functional problem solution program PRFZ
7. Repeated problem solution subprogram PPRZ
8. Diagnostic problem solution subprogram PRDZ
9. Computer disconnect subprogram POM
10. Repair
11. Malfunction
12. Correct
13. Failure
14. Incorrect
15. UVM_2 correct
16. Simplex operating mode SRR

$USpM_2$ are used. When the results coincide, they are sent out into external circuits along which they go (for example) to the controlled object. The system's functioning in this mode is called operation in the asynchronous duplex mode. If the results do not coincide, actions are performed to determine in which of the computers there was a failure or malfunction. As a rule, a malfunction is detected by the method of repeated computation, using the same initial data. If a failure occurs, diagnostic tests are run to determine which computer failed and the point at which the defect appeared. During the time when the computer that failed is being repaired, the system operates without it. This mode of system operation is called the simplex mode. Figure 5-4 is a temporal diagram of the operation of a redundant system consisting of two computers and that realizes the operating mode described above. After the command "Start" is received, initial start subprogram PNP is turned on in each computer, whereupon it checks to see that the computer's units are in good working order and that the computer is ready to function. This same subprogram prepares the computer to solve functional problems. Functional problem solution program PRFZ is turned on after a signal that the computer's units are ready to function is received from the PNP. When a signal that the computer is malfunctioning is received, repeated problem solution subprogram PPRZ is turned on. If the malfunction was a random one, problem solution continues after repeated solution of the entire problem or part of it. Duplex operating mode DRR is realized in the system. If a failure occurred in the system, diagnostic problem solution subprogram PRDZ is turned on in each computer. Which of the computers is out of order is determined on the basis of the functioning of the diagnostic programs. Computer disconnect subprogram POM disconnects the defective computer from the system so it can be repaired, while the computer that is in good working order begins to operate and continues solving functional problems in the simplex mode.

The operational reliability of a redundant system of this type can be evaluated in the following manner. The UVM's belong to the class of extended-use, multiple-action, repairable systems. In such systems, as is well known, along with sudden failures and malfunctions it is possible for gradual failures to occur, and the manifestation of such failures in multielement systems is extremely complicated. As a rule, in a UVM there are provisions for periodic preventive maintenance, during which elements in a "critical" state are detected. These elements are replaced with new ones, as a result of which the evaluation of UVM reliability is usually limited to a discussion of sudden failures only.

Considering the remarks previously made relative to the nature of a stream of failures, let the operational reliability of an unmonitored computer be described by the following relationship:

$$P_e(t) = K_r \exp\{-t/T_0\}. \tag{5-7}$$

Let us designate the degree of multiplicity of redundance (the number of reserve devices) as m_{red}, while the probability of failureless operation of a unit with a backup is $P_{red}(t)$. In the first approximation, the probability of failureless operation of a repairable system that is backed up with an equivalent reserve can be evaluated with the formula

$$P_{1red}(t) = 1 - [1 - P_e(t)]^{m_{red}+1}, \tag{5-8}$$

where $P_{1red}(t)$ = probability of failureless operation of a backed-up unit, using permanent redundancy.

Let it be required that a necessary level of reliability P_3 of the backed-up system be insured; that is, that

$$P_{1red}(t) \geq P_3. \tag{5-9}$$

Let $P_{1red}(t) = P_3$.

Substituting the value of $P_e(t)$ from (5-8) into (5-7) and taking (5-9) into consideration, after making the transformation we obtain

$$e^{-t/T_0} = (1 - \sqrt[m_{red}+1]{1 - P_3})(1 + T_{rep}/T_0). \tag{5-10}$$

If we expand e^{-t/T_0} into a series and take the first two terms of this series into consideration, the allowable value of the average time of failureless operation T_0 of a computer can be evaluated in the following manner:

$$T_0 \approx [(t + T_{rep})\sqrt[m_{red}+1]{1-P_3}] - T_{rep}. \qquad (5\text{-}11)$$

A more accurate value of T_0 can be obtained by solving equation (5-10) graphically. In order to do this, we introduce the following definitions [10]:

$$f_1(T_0) = e^{-t/T_0};\ f_2(T_0) = (1 + T_{rep}\ \ T_0)\left(1 - \sqrt[m_{red}+1]{1-P_3}\right).$$

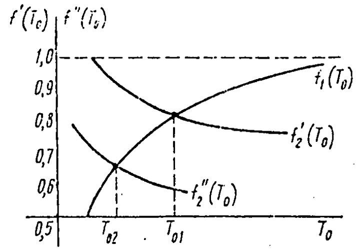


Figure 5-5. Toward determining the average time of failureless operation.

Figure 5-5 depicts the graphic solution of transcendental equation (5-10) for two values of $f_2(T_0)$, it being the case that $f_2'(T_0) > f_2''(T_0)$.

The formulas for evaluating the reliability of a backed-up system that were presented above are correct only for a system in which a part that fails is repaired immediately after the failure occurs. If it is not done until after the entire system goes out of operation, the effectiveness of the redundancy drops substantially.

In order to deal with elements that fail immediately after they stop functioning, it is necessary to have special indicators, the installation of which is not always possible. As was mentioned earlier, periodic preventive correction of a system's reliable properties is used for the purpose of increasing the effectiveness of system redundancy. In connection with this, the preventive maintenance period T_{prv} should be less than the average time of failureless operation of one of the j-th subsystems into which the system is divided [41]; that is, $T_{prv} < T_{0j}$.

Redundancy by replacement assumes that the computers are categorized as basic and reserve. Under normal conditions, only the basic computer operates. It carries out all the system's functional problem solution algorithms and communicates with the external equipment (the data transmission equipment) and the devices for depicting information on the operator's console. The reserve computer can be in one of the following modes: a) turned off (cold, unmonitored reserve), b) waiting (hot, unmonitored reserve), c) check monitoring (the hot, monitored mode), d) duplicated storage of the most important information produced by the basic computer's operation.

In order to reduce time losses at the reserve unit's input, the last two modes are those primarily used in control systems. Let us discuss the realization methods and the special operating features of a backed-up system operating with duplicated storage of the most important information. This mode insures the most rapid engagement of the reserve unit.

In order to insure continuity of control in case of failure of the basic computer, certain information must always be present in the reserve unit. This information is kept in its internal storage and reflects the course of the computational process at the given moment, the basic results of the calculations, the state of the controlled object(s), the control decisions that were made earlier and are important from the viewpoint of continued operation, and the technical state of the individual units that make up the control system. In general form, Figure 5-6 is a structural diagram of a duplicated system with redundancy by replacement. The special feature of this system is the presence of switching devices at the input P_{in} and the output P_{out}, as well as in the circuits for exchanging information between computers POI.

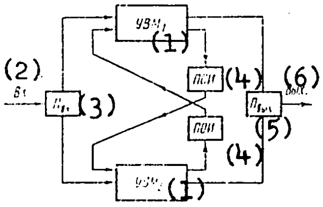


Figure 5-6. Structural diagram of duplicated system with redundancy by replacement.

Key:
1. UVM
2. Input
3. P_{in}
4. Circuits for exchanging information between computers
5. P_{out}
6. Output

When this redundancy method is used, the requirements for reliability of the switching devices are extremely rigorous, since the failure of any one of them causes the entire system to fail.

If, in the first approximation, we ignore failures of the backup information switching devices, the probability of failureless operation of a backed-up system with redundancy by replacement can be determined in the following manner:

$$P_{2red}(t) = P_{in}(t)\,P_{out}(t)\,\{1-[1-P_c(t)]^{m_{red}+1}\}, \qquad (5\text{-}12)$$

where $P_{in}(t)$, $P_{out}(t)$ = probability of failureless operation of the switches at the system's input and output.

As for the case of permanent redundancy, let $P_{2red}(t) = P_3$ and $P_e(t) = K_r e^{-t/T_0}$. Substituting these values of $P_{2red}(t)$ and $P_e(t)$ into (5-12), we can determine the average time of faileless operation of a single computer:

$$T_0 \approx \frac{t + T_{rep}}{\sqrt[m_{red}+1]{1 - P_3/[P_{in}(t)P_{out}(t)]}} - T_{rep} . \qquad (5\text{-}13)$$

For a duplicated system, $m_{red} = 1$, so that

$$T_0 \approx \frac{t + T_{rep}}{\sqrt{1 - P_3/[P_{in}(t)P_{out}(t)]}} - T_{rep} .$$

These formulas -- as was the case for permanent redundancy -- are correct for systems with re-establishment of the reliability properties of the system's elements either immediately after a failure occurs or during preventive maintenance.

An essential feature of the organization of redundancy by replacement for systems consisting of two or more computers is the necessity of exchanging back-up information between the basic and reserve computers. This feature causes the operating programs of each of the computers in the complex to be constructured in such a manner that in addition to solving the basic functional problems, performance of the following operations is insured: 1) initial input of the information into the reserve UVM's memory and the starting of its program in the check monitoring mode; 2) exchange of data on the state of the basic and reserve computers' equipment during operation; 3) periodic forwarding of back-up information from the main computer's memory to that of the reserve one; 4) switching of the UVM's when an irregularity arises in the basic computer or on command of the operator; 5) switching back from the reserve UVM to the basic one when the irregularity is eliminated without, for all practical purposes, interrupting the continuity of control.

Depending on the state of the reserve computer, the following modes are distinguished: 1) duplex, when the basic and reserve computers are both in good working order and are both turned on; 2) duty, when one of the computers is turned off or undergoing repair.

Let us discuss the principles of the program realization of the duplex and duty operating modes of the redundant system.

It is possible to realize the duplex mode either by simultaneously starting two computers that are in good working order and were not previously in operation, or by connecting the reserve

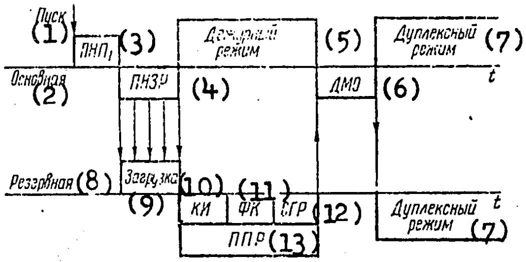


Figure 5-7. Temporal diagram of engagement of two computers in the duplex mode of operation.

Key:

1. Start
2. Basic
3. Initial start subprogram PNP
4. Reserve computer initial load subprogram PNZR
5. Duty mode
6. Intercomputer exchange supervision subprogram DMO
7. Duplex mode
8. Reserve
9. Loading
10. Information checking and correction KI
11. Reserve computer readiness check FK
12. Ready signal SGR
13. Special start subprogram PPR

computer to the already functioning basic one. Depending on the original state of the computers, the duplex mode organization algorithms will be substantially different.

Simultaneous starting of the two computers is carried out in the following manner. The available computers are designated as basic and reserve. The operation of the basic computer begins with the running of initial start subprogram PNP$_1$ (Figure 5-7), which enters all the initial information in the computer's internal memory and turns on special reserve computer initial load subprogram PNZR. With the help of this subprogram, all the initial information is shared with the reserve computer's internal memory. After loading, the reserve computer is ready for operation and is turned on by special start subprogram PPR. The functions of this subprogram include checking and correcting the information receved from the basic computer (KI), checking the good working order of the reserve computer's equipment (FK), and issuing the results of this check (readiness for operation -- or lack of it -- of the reserve computer). After receiving ready signal SGR, the basic computer's operation is interrupted and intercomputer exchange supervision subprogram DMO is turned on. The end of this subprogram's operations signals the completion of the system's conversion to the duplex mode.

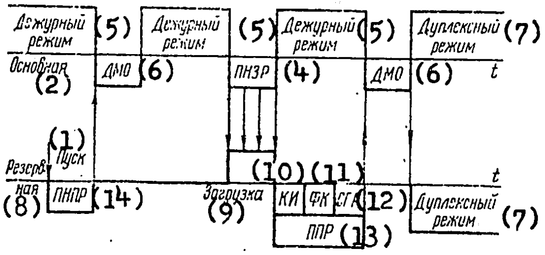


Figure 5-8. Temporal diagram of connection of reserve computer.

Key:

1,2,4-13. See Figure 5-7

14. Reserve computer initial start subprogram PNPR

Figure 5-7 is a temporal diagram of the conversion of the computers into the duplex mode of operation.

A reserve computer is connected to an already operating basic computer in the following manner. Upon receiving a "Start" signal, reserve computer initial start subprogram PNPR is turned on. This subprogram clears the reserve computer's internal memory and registers, makes the initial adjustment of its subordinate peripheral equipment, and issues an inquiry about connecting it to the basic computer. When this inquiry is received, the implementation of the functional programs by the basic computer is interrupted and the intercomputer exchange supervision subprogram is turned on. This subprogram prepares the PNZR to forward information to the reserve computer. In order that the transferred information not be obsolete, the basic computer operates in the duty mode for some time before the reserve computer is loaded. After the information is forwarded, the working sequence of the basic and reserve computers is analogous to the one described earlier. Figure 5-8 is a temporal diagram of the connection of the reserve computer.

The exchange of information between the computers includes: 1) an exchange of interaction signals over special trunks; 2) the forwarding of numerical information to one of the computers.

The interaction signals are preparatory signals that insure the synchronization of the operation of the basic and reserve computers before the forwarding of the numerical information. The duty mode is realized most frequently when an irregularity arises in one of the computers.

Let us examine the complex's operation when an irregularity appears in one of the computers. In this situation, it is

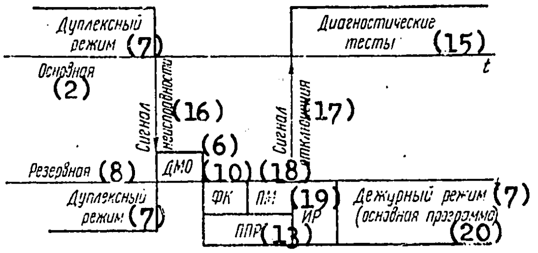


Figure 5-9. Temporal diagram of process of switching computers from duplex to duty mode.

Key:
2, 6-8, 10, 13. See Figure 5-7
15. Diagnostic tests
16. Defect signal
17. Disconnect signal
18. Computer switching subprogram PM
19. Mode change subprogram IR
20. Basic program

necessary to disconnect the basic computer and shift the reserve computer to the duty mode for the implementation of the functional programs. The computer that is out of order must be switched to the defect search mode with the help of diagnostic programs or shut down so that a manual search for the irregularity can be made. A defect signal received from the monitoring units over the interaction signal transmission trunks causes the reserve computer's operation to be interrupted and turns on intercomputer exchange supervision subprogram DMO. This program, in turn, on reserve computer monitoring subprogram FK, which checks the readiness for operation of that computer's units, and then computer switching subprogram PM. This subprogram issues a signal to change the defective computer into the defect search mode and turns on mode change subprogram IR, which switches the reserve computer into the duty mode for the implementation of the control system's functional programs using the available reserve information. Figure 5-9 is a temporal diagram of the process of switching the computers from the duplex to the duty mode.

Since the complex is composed of identical machines, redundancy by replacement can be organized by the mixed redundancy method as well as by total redundancy. For example, peripheral gear redundancy can be organized along with computer redundancy. In order to do this, the gear must be fitted with a monitoring system. When a defect is detected in the peripheral gear of one of the computers, it is sufficient to switch only the peripheral gear instead of the entire computer. Figure 5-10 is a temporal diagram of the process of switching the peripheral gear when a defect is discovered in the basic computer's peripheral gear. As a result of this switching, the basic

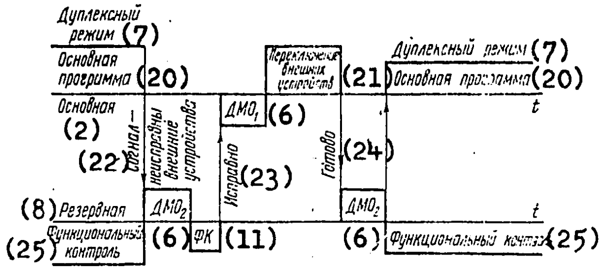


Figure 5-10. Temporal diagram of process of switching peripheral gear.

Key:

2, 6-8, 11, 20. See Figures 5-7 and 5-9

21. Switching of peripheral gear

22. Peripheral gear defect signal

23. In good working order

24. Ready

25. Functional monitoring

computer will operate through the reserve computer's coupling unit. The basic computer's nonworking peripheral gear is disconnected and a search is made for the defects in these units.

5.4. Use of Automatic Monitoring Methods to Increase the Operational Reliability of UVM's

The equipment redundancy methods discussed in Section 5.3 increase the probability of failureless operation of a control system. In order to increase the probability of receiving correct information at the control system's output, various methods of monitoring its operation are used.

A set of facilities for automatically monitoring the course of the computational process and the equipment's fitness for operation is called a monitoring system.

UVM monitoring systems are given the tasks of automatically detecting equipment malfunctions and failures with a minimum time lag relative to the moment of appearance of such defects; correcting errors caused by random malfunctions or eliminating whatever effect they have; determining the point of failure in the computer. In control systems, in many cases malfunctions in the operation of the control unit are as unallowable as failures, so increasing the reliability of the computation results is especially important for UVM's.

The classification of the existing automatic monitoring methods that are used in UVM's is shown in Figure 5-11. Regardless of

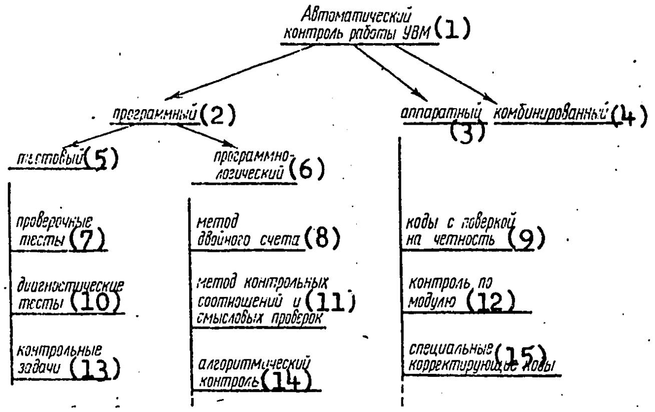

Автоматический контроль работы УВМ (1)
программный (2)
аппаратный (3)
комбинированный (4)
тестовый (5)
программно-логический (6)
проверочные тесты (7)
метод двойного счета (8)
коды с поверкой на четность (9)
диагностические тесты (10)
метод контрольных соотношений и смысловых проверок (11)
контроль по модулю (12)
контрольные задачи (13)
алгоритмический контроль (14)
специальные корректирующие коды (15)

Figure 5-11. Classification of UVM monitoring methods.

Key:

1. Automatic monitoring of UVM operation
2. Programmed
3. Built-in equipment
4. Combined
5. Testing
6. Program and logic
7. Check tests
8. Binary counting method
9. Codes with parity checks
10. Diagnostic tests
11. Method of monitoring relationships and meaning checks
12. Modulus monitoring
13. Check problems
14. Algorithmic monitoring
15. Special corrective codes

the type of monitoring, they are all constructed on the principle of using redundancy -- temporary, informational, or equipment. Therefore, the introduction of monitoring cannot increase the probability of errorless computer operation. Monitoring makes it possible to increase only the reliability of the issued information, since with its help all or some part of the unreliable information is not passed to the computer's output. In connection with this there is a reduction in the number of solutions passed into the system from the computer per unit of time.

Let us discuss the organizational principles and the areas of application of the various methods for monitoring UVM operation.

The Programmed Monitoring Method. This type of monitoring is based on the principle of using special programs for monitoring both individual units and the computer as a whole. The basic advantage of this method is that its organization does not require special equipment; the disadvantage is that more time is consumed during the monitoring process and additional memory capacity is needed to store the monitoring programs.

Programmed control is categorized as being either program and logic or testing.

Test monitoring is the monitoring of the fitness for work of a UVM and its separate assemblies with the help of testing programs. The purpose of test monitoring is to detect failures and search for the defective assembly or element. Test monitoring is done periodically, at times when the functional programs are not running. Therefore, it cannot be used to detect malfunctions.

Test monitoring is based on the solution of several problems with given initial data and the comparison of the results with standards. If the test problem's solution does not coincide with the standard solution, it is concluded that either the computer or one or more of its separate units is not in good working order. Depending on the purpose, the following varieties of tests are used: check, diagnostic and check problems.

Check tests are supposed to detect defects in the computer, allowing for the specific nature of the computer's logic structure and elements. Each type of industrially produced computer has a set of check tests. The tests are constructed in such a fashion that the unit (assembly) being tested is checked under the most rigorous conditions, while the other units are, contrarily, operating under the easiest possible conditions. A set of testing programs usually has check tests for the arithmetic unit, the internal and external memories, the control unit, the system for communicating with the controlled object, and so forth. Check tests are turned on before a computer or a unit begins to operate.

Diagnostic tests are used to determine the location of an irregularity. They are turned on after information about a computer failure is received and are realized with the help of a human or another computer. The general software for a computer also includes a set of diagnostic testing programs.

Check problems are used for systematic programmed checking of a computer. They can be engaged periodically, on the operator's initiative, or automatically, during the basic problem solution process. In the latter case, programs for solving the check problems are entered in the computer's memory and are run at certain intervals. The decision on the correctness of a computer's operation is made on the basis of the results of a comparison of the solution of the check problem with a standard (the correct answer). Check problems must be formulated so that they provide a complete check while using little time and little memory volume for the realization of the check problem solution program. For series-produced computers, a set of recommended check problems is also part of the general software system.

Program and logic monitoring is intended to discover random malfunctions in a computer's operating process and to suppress the output of an incorrect result. The types of program and logic monitoring most widely used in UVM's are: monitoring by the binary counting method, the monitoring relationships and meaning checks method, and algorithmic monitoring.

The binary counting method checks a problem solution for the absence of malfunctions according to the following algorithm:
1. $P\{A_1 = A_2\}$ = result correct, transfer to 3; additional solution of A_3 and transfer to 2.
2. $P\{A_1 = A_3 \vee A_2 = A_3\}$ = result is A_3 and transfer to 3; transfer to 4.
3. Next computation step, problem is being solved correctly.
4. Stop -- irregularity in the computer.

Here A_1, A_2 and A_3 are the results of the first, second and third solutions of the problem given the same initial data.

Thus, when monitoring by the binary counting method is used, it is possible to have a triple execution of the same section of a program or of the entire program. Therefore, this method is also called the binary-ternary counting method. The organization of the binary counting method is extraordinarily simple and does not require additional equipment. Even when malfunctions are absent, however, the problem solution time is doub ed. When malfunctions appear, it is increased even more. When this method is used, systematic errors that are the result of failures or other causes are not detected.

The method of monitoring relationships and meaning checks uses an indirect check of a computer's functioning by computing several known relationships. For example, in computations of the function sin α, the correctness of the computation can be checked by computing the relationship $\sin^2 \alpha + \cos^2 \alpha = 1$. The basic advantage of the monitoring relationships method is the possibility of discovering systematic errors with its help. As a rule, it is used in combination with the binary counting method.

Algorithmic monitoring is based on the use of truncated basic problem solution algorithms (a truncated algorithm is an approximate basic problem solution algorithm that is characterized by a shorter realization time in comparison with the complete algorithm). If the solutions produced by the basic and truncated algorithms differ insignificantly, it is decided that the computer is functioning correctly. For example, the exact algorithm for solving differential equations utilizes the (Runge-Kutt) method, while the truncated algorithm for solving the same equations uses Euler's method.

Algorithmic monitoring makes it possible to check a solution for the absence of gross errors. Its basic advantage is less consumption of computer time for the monitoring process in comparison with (for example) the binary counting method; the disadvantages of this method are its limited application, since it is not always possible to construct a truncated algorithm, and the impossibility of complete detection of random malfunctions.

Another variety of program and logic monitoring is the method of check sums. This method is mainly used to check the correctness of information storage in a computer's memory. The essence of the check sums method is as follows. Let the transmission of some mass of numbers be monitored. In connection with this, the sum of all the numbers in the mass has been previously calculated. Counting is usually done modulo 2^{n-1} (n = = length of the numbers). After the transmission is completed, the check numbers are added and the result is compared with the preliminarily computed check sum.

The Built-In Equipment Monitoring Method. This method is based on the addition to a computer of special equipment that continuously monitors the operation of the computer or its units during the problem solution process. Such a monitoring system construction does not reduce the computer's productivity for all practical purposes, although it can increase the amount of equipment substantially because of the inclusion of the monitoring device's in its composition. Built-in equipment monitoring systems are used in conjunction with special redundancy codes, which makes it possible not only to detect errors but also to correct them (when corrective codes are used).

Methods of monitoring codes for parity and variants of the method of absolute value monitoring of numbers are most widely used in built-in equipment monitoring systems [55].

The use of one built-in equipment monitoring method or another is decided upon during the development of the computer, and when it is used in a control system with series-produced UVM's, the choice of method is not left up to the system's developer. For such computers, the built-in equipment monitoring system is considered to be already given.

Combined Monitoring Methods. These monitoring methods are based on a combination of programmed and equipment methods and make it possible to reduce both the amount of monitoring equipment and the monitoring time. In such systems, error detection is usually the province of the equipment facilities, while the nature of the malfunction (random or systematic) is analyzed by the programmed monitoring method. For example, a random malfunction can be corrected by using the binary counting method. In this case, the time consumed by the monitoring process is reduced because binary counting is performed only when a malfunction is discovered. In some computers, the combined monitoring method is realized in the form of program and logic monitoring. This method specifies that the monitoring devices, which were added to the computer when it was being developed, are not turned on continuously, but only periodically when a defect in the computer's functioning is detected by one of the programmed methods. Other varieties of the combined monitoring method are also possible. The inclusion in the computer of special monitoring circuits is characteristic of all variants of monitoring by the combined method.

Table 5-1

Характеристики систем контроля(1)	Контроль вычислений(2)		
	программно-логический(3)	тестовый (4)	аппаратный (5)
Время, затрачиваемое на контроль (% ко времени решения задачи)(6)	100 и более (7)	10÷15	0
Дополнительное оборудование (% к основному оборудованию)(8)	0	0	15÷25

Key:

1. Characteristics of monitoring systems
2. Computation monitoring
3. Program and logic
4. Testing
5. Equipment
6. Time consumed for monitoring (percent of problem solution time)
7. 100 and more
8. Additional equipment (percent of basic equipment)

The monitoring of the operation of any device, including a UVM, is accomplished by introducing different types of redundancy that, in the end, result in enlarging the amount of computer equipment or the problem solution time. Approximate average values of the additional time and amount of monitoring equipment for the monitoring of computation correctness are presented in Table 5-1 for different types of monitoring [12].

We should also mention here the work that is being done to use the principles of adaptation not only to improve the basic functional characteristics of control systems with UVM's, but also to improve the reliability characteristics of such systems [39,53]. At the present time this area is in the process of formulation, and requires the performance of additional work in order to derive engineering techniques for calculating the reliability of specific control systems.

Selecting the Monitoring System. The effectiveness of a monitoring system is evaluated by the coefficient of monitoring quality $K_k(t)$, which is the probability of obtaining an errorless answer after monitored operation of the device (computer) for time t.

By definition,

$$K_{\text{к}}(t) = 1 - P_{er}(t), \qquad (5\text{-}14)$$

Here

$$P_{er} = qP(\bar{A}) \qquad (5\text{-}15)$$

is the probability that the monitoring system will pass errors through [36]. $P(\bar{A})$ = probability of the appearance of errors in checked and monitored devices ($\bar{A}$ = an event consisting of the monitored operation of a device (ccmputer) with errors), q = probability of the passage of errors by the monitoring system.

After substituting (5-15) into (5-14), we have

$$K_{\text{к}}(t) = 1 - qP(\bar{A}). \qquad (5\text{-}16)$$

If we allow for the appearance of several errors simultaneously; that is, the appearance of errors with different degrees of multiplicity, then

$$P_{er} = q_1P(\bar{A}_1) + q_2P(\bar{A}_2) + \ldots + q_{n+n_{\text{K}}}P(\bar{A}_{n+n_{\text{K}}}) = \sum_{i=1}^{n+n_{\text{K}}} q_iP(\bar{A}_i),$$

$$K_{\text{к}}(t) = 1 - \sum_{i=1}^{n+n_{\text{K}}} q_iP(\bar{A}_i),$$

where i = degree of multiplicity of the error; n = number of information representation bits; n_k = number of check bits ($n_k = 0,1,2,\ldots$).

The values of $P(\bar{A}_i)$ and q depend on the monitoring method that is used and the equipment realization of the device (computer) as a whole. The technique for determining them is described in sufficient detail in [36] and will not be presented here.

Monitoring quality coefficient K_k characterizes the functional reliability of the output of information by a device (computer) that is completely monitored. Actually, only part of the equipment or the computations can be checked by the monitoring system. For example, when two computers operating in the duplex mode and solving the same problem are monitored, the operation of the result comparison circuit (see Figure 5-3) is not monitored. For such devices, functional reliability -- the probability of obtaining correct information at the device's output -- will be determined by the quality coefficient of the monitored units and the probability of failureless operation of the equipment not covered by the monitoring system:

$$P_f(t) = K_к(t) P_{nm}(t), \qquad (5\text{-}17)$$

where P_{nm} = probability of failureless operation of the units that are not monitored.

In addition to increasing the probability of the output of a correct results, some of the types of monitoring -- primarily equipment and test monitoring -- also make it possible to reduce computer repair time when one of its individual units fails.

This last advantage is achieved by reducing the amount of time it takes to search for the defect. Thus, in some cases the introduction of monitoring makes it possible to improve such a parameter of computer operational reliability as readiness factor K_r.

If we designate the average time between failures and the average repair time of a monitored device (computer) as T_{om} and $T_{rep\ m}$, respectively, then the readiness factor K_{rm} (which allows or monitoring) can be defined as

$$K_{rm} = T_{om}/(T_{om} + T_{rep\ m}). \qquad (5\text{-}18)$$

For systems with monitoring, the fulfillment of the following inequalities is typical:

$$T_{om} \leqslant T_o; \; T_{rep\ m} \leqslant T_{rep}. \qquad (5\text{-}19)$$

The first inequality is caused by the introduction of the additional monitoring equipment; the second, by the monitoring system's property of localizing sources of systematic errors (failures) when the computer is operating, thereby reducing the defect search time.

The operational reliability of computers with monitoring $P_{em}(t)$ is determined with a formula that is analogous to the formula for determining the operational reliability of a computer without monitoring:

$$P_{em}(t) = K_{rm} P_f(t) = K_{rm} K_к(t) P_{nm}(t). \qquad (5\text{-}20)$$

In order to select a monitoring system, it is usually necessary to insure the fulfillment of an inequality of the type

$$P_f(t) \geqslant P_{fg}(t);\ K_{rm} \geqslant K_{rg}, \qquad (5\text{-}21)$$

where P_{fg}, K_{rg} = respectively, the given probability of obtaining a correct solution (given reliability of obtaining a result) and the device's (computer's) given readiness factor.

If several monitoring systems satisfy requirement (5-21), the system providing the maximum operational reliability value is considered to be the optimum one.

Thus, when condition (5-21) is fulfilled, the monitoring system is chosen according to the criterion

$$J_l = \max_l \{P_{el}(t)\},\ l = 1,\ 2,\ \ldots,\ g, \qquad (5\text{-}22)$$

where g = number of compared monitoring systems; $P_{e\ell}(t)$ = operational reliability of a computer with the control system designated by number ℓ.

In the class of programmed monitoring methods, in which the choice of monitoring systems for ready UVM's intended for operation in control systems for production processes is made, an increase in the reliability of the information obtained is achieved by increasing the solution time. In connection with this, the possibility of introducing a specific monitoring system must be solved not only with due consideration for inequalities (5-21) and criterion (5-22), but also for insuring the given cycle of operations in formulating the control actions. In connection with this, it is required that

$$T'_{tot} + T_\kappa \leqslant T_c, \qquad (5\text{-}23)$$

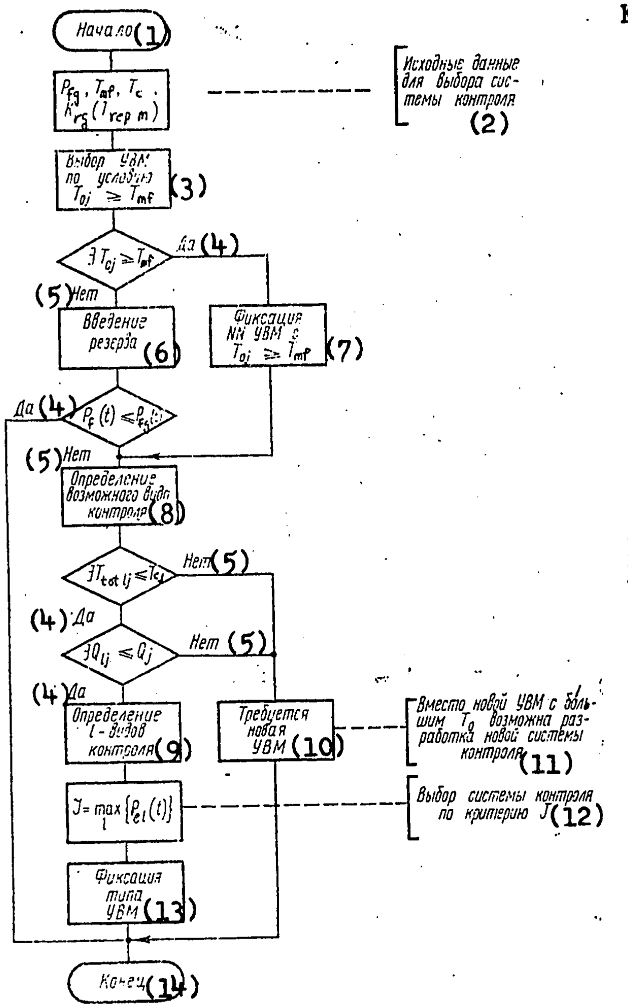
where T'_{tot} = problem solution time without allowing for time spent in monitoring; T_k = time consumed for monitoring.

For a given problem, the value of T_k depends on the adopted monitoring method and can be determined by methods described in Section 4.7. In the first approximation, it can be taken into account as a percentage of the total solution time T_{tot}; that is, $T_k = \varkappa T_{tot}$.

For monitoring that is temporally nonintegrated with the implementation of the basic assignments, the percentage of the total solution time that is taken for monitoring in order to insure the required reliability in obtaining results can be evaluated with the empirical formula [12]

$$\varkappa = -\frac{\ln P(t)}{k} \cdot \frac{P_f}{1 - P_f},$$

where P(t) = probability of failureless operation during time t;


Начало (1)
P_{fg}, T_{mf}, T_c, k_{rg} ($T_{rep\ m}$)
Исходные данные для выбора системы контроля (2)
Выбор УВМ по условию $T_{0j} \geq T_{mf}$ (3)
$\exists T_{0j} \geq T_{mf}$
Да (4)
(5) Нет
Введение резерва (6)
Фиксация NN УВМ с $T_{0j} \geq T_{mf}$ (7)
Да (4)
$P_f(t) \leq P_{fg}(t)$
(5) Нет
Определение возможного вида контроля (8)
$\exists T_{tot\ lj} \leq T_{cj}$
Нет (5)
(4) Да
$\exists Q_{lj} \leq Q_j$
Нет (5)
(4) Да
Определение l-видов контроля (9)
Требуется новая УВМ (10)
Вместо новой УВМ с большим T_0 возможна разработка новой системы контроля (11)
$J = \max_l \{P_{el}(t)\}$
Выбор системы контроля по критерию J (12)
Фиксация типа УВМ (13)
Конец (14)

Key:
1. Beginning
2. Initial data for selecting monitoring system
3. Selection of UVM according to condition $T_{0j} \geq T_{mf}$
4. Yes
5. No
6. Introduction of redundancy
7. Fixation of NN of UVM with $T_{0j} \geq T_{mf}$
8. Determination of possible type of monitoring
9. Determination of ℓ-types of monitoring
10. New UVM required
11. Instead of new UVM with larger T_0, it is possible to develop a new monitoring system
12. Selection of monitoring system according to criterion J
13. Fixation of type of UVM
14. End

Figure 5-12. Algorithm for selecting monitoring system.

P_f = required reliability of obtaining results; k = normalizing coefficient.

In order to evaluate the required monitoring time, k can be taken as approximately equal to unity. From that,

$$\varkappa \approx -\ln P(t)\, P_f/(1-P_f). \qquad (5\text{-}24)$$

Thus, the technique for selecting a system to monitor the operation of finished UVM's used in control systems can be as follows (Figure 5-12).

1. The following should be given as the initial data: a) P_{fg} = = reliability of the obtained result; b) T_{mf} = mean time between failures; c) K_{rg} = readiness factor (or repair time $T_{rep\ m}$). If the reliability of the obtained result is not given, the probability of failureless operation P(t) can be used as P_{fg} [36].

2. For the selected type of UVM (see Chapter 4), the necessity of introducing redundancy is determined on the basis of an analysis of the inequality

$$T_{0j} \leqslant T_{mf}, \qquad (5\text{-}25)$$

where T_{0j} = mean time between failures of the j-th UVM from the group of computers satisfying the other selection conditions (see Chapter 4).

If condition (5-25) is not fulfilled, it is necessary to introduce redundancy into the system; the type of redundancy is determined both by the necessity of observing inequality (5-25) and the special requirements that may be formulated in each specific case.

3. On the basis of an analysis of the values of $P_f(t)$ and K_r for the specific UVM chosen for use in a control system, the question of the necessity of introducing a monitoring system is asked (see condition (5-21)).

4. If condition (5-21) is not fulfilled, the monitoring systems (or system) that can possibly be used in the UVM are determined in the class of programmed monitoring methods, with due consideration for the possible limitations caused by the specific nature of the control system's operation.

5. For each UVM chosen according to other technical indices for possible use in the control system under discussion, and for each of the previously determined possible monitoring systems, the values of $K_k(t)$, $P_{nm}(t)$ and K_{rm} are found and the fulfillment of condition (5-21) -- allowing for monitoring -- is checked.

6. If, as a result of the introduction of a monitoring system, condition (5-21) is fulfilled when several monitoring systems are used, further selection is made with due consideration for the fulfillment of inequalities

$$T_{tot\ ij} \leqslant T_c;\ Q_{ij} \leqslant Q_j^{M}, \qquad (5\text{-}26)$$

where $T_{tot\ \ell j} = (T'_{tot\ j} + T_{k\ell})_j$ = problem solution time, allowing for the introduction of the ℓ-th monitoring system for the j-th UVM; $Q_{\ell j} = (Q + Q_{k\ell})_j$ = required memory volume, allowing for the ℓ-th monitoring system for the j-th UVM; Q^m_j = memory volume allocated to the j-th UVM for problem solution.

7. If condition (5-26) is fulfilled for several monitoring systems, the choice of the monitoring system for the j-th UVM is made according to criterion (5-22).

The engineering method of selecting a monitoring system that was presented above is only part of the general UVM selection technique for realizing given control algorithms with the help of series-produced UVM's for general industrial purposes (see Chapter 4).

As with the selection of a UVM, the problem of choosing a monitoring system is solved in stages, by the method of successive approximations, which insures the fulfillment by the system of the given reliability characteristics. In connection with this, the reliability requirements for the control unit (UVM) can be determined from the condition of insuring the optimum level of reliability for the given level of reliability of the object and the units that link it to the computer (see Section 5.2). In this case, after the requirements for the control unit's level of reliability are determined, the monitoring system selection technique explained above is retained completely.

5.5. Effect of Reliability Indices on the Structure of an Automatic Control System With a UVM

In the automation of technological processes and objects with the help of UVM's, the two most widely used types of control system structure formulation are centralized and hierarchical (Figure 5-13). In the first case (Figure 5-13a), the UVM receives information x_i from controlled objects O_i, monitors their mode, and in accordance with the control goals works out control actions u_i, which act on the objects through actuating elements IO_i. As a result of the UVM's high operating speed and large logic capabilities, the number of control circuits n can be extremely large. The use of such a structure means that with the help of a single UVM controlling unit, the problems involved in controlling many objects can be solved.

Centralization of control makes great demands on the computer's operational reliability, because if it goes out of order it disrupts the control of an entire technological process or even a production complex. The latter always leads to great economic losses.

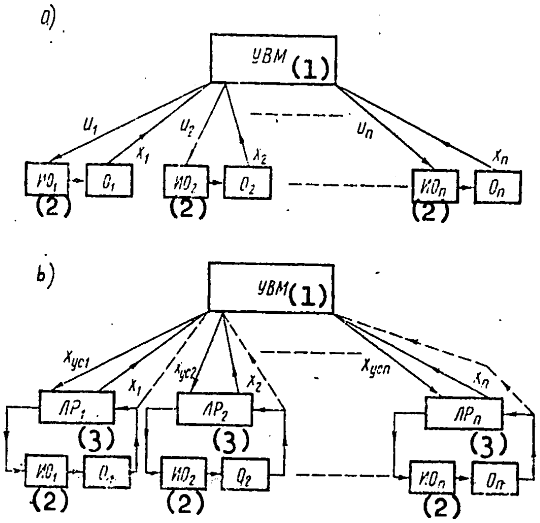


Figure 5-13. Structures of control systems with UVM's.

Key:

1. UVM
2. Actuating element IO
3. Local regulator IR

A two-level hierarchical system (Figure 5-13b) normally uses local automatic units -- local regulators LR$_i$ -- on the lower level and a coordinating UVM on the upper level. In such a structure, direct control is exercised by local regulators LR$_i$. The UVM coordinates their operation, optimizing the control process as a whole. The UVM affects the process indirectly, by changing the settings $x_{yc\ i}$ of the local regulators. In a hierarchical system the requirements for the UVM's operational reliability can be relaxed, because control does not cease if it goes out of order, but merely becomes nonoptimal. The costs for this system are greater than for a centralized one, since it is necessary to install local regulators in it. Besides this, each such regulator will have limited reliability, which will also lead to losses if the regulator goes out of order.

It can be assumed that when there is a large number of local regulators, the increase in costs for installing them does not compensate for the losses arising when a UVM in a centralized system goes out of order and during the time needed to repair it. Let us discuss the conditions where it is advisable to use centralized and hierarchical structures in automatic control systems with UVM's [41]. We will make our evaluation according to the criterion of complete expenditures. In order to simplify the analysis, let us assume that all n control circuits are identical, the expenses for operating the systems are also identical, and the unreliability of the actuating elements IO_i can be ignored. The system operates cyclically; the duration of the cycle is constant and equals T_c.

We will use the following definitions: $X = \{x_1, x_2, \ldots, x_n\} =$ = vector of the controlled object's state; $U = \{u_1, u_2, \ldots, u_n\} =$ = control vector; $X_{yc} = \{x_{1yc}, x_{2yc}, \ldots, x_{nyc}\}$ = vector of settings (given values); $B^* = \varphi(X, U^*)$ = optimum value of the control functional (control efficiency); U^* = optimum value of the control vector; $B = \varphi(X, U)$ = current control efficiency value.

When a UVM in a centralized system goes out of order, the following situations are possible:

1. Immediate cessation of the controlled object's functioning; losses w_0 are determined by the object's downtime and can be evaluated as a complete loss of effectiveness during the time T_{rep} required to eliminate the defect:

$$w_0 = B^* T_{rep} \cdot$$

2. The system continues to function normally for some time after the UVM's failure and only then is there disruption of its operation; in this case the object's behavior can change, depending on the position of the actuating elements IO_i: a) they remain in the same position they occupied at the moment of the emergency t_0; that is, $U = U_0$; b) they change to an emergency state characterized by the value $U = U_{em}$.

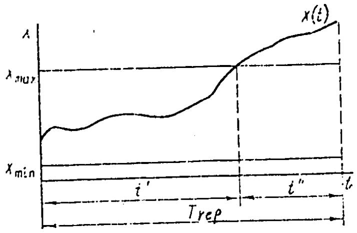


Figure 5-14. Toward determining the loss when a UVM fails.

In the second situation, loss w_1 is composed of two parts (Figure 5-14): in section t' it is proportional to the difference between the optimum and actual efficiency values; that is,

$$w_{11} = (B^* - B_1)\, t'. \qquad (5\text{-}27)$$

In section t", parameter $x_i(t)$ goes beyond its allowable limits -- $x_i(t) > x_{i\ max}$ -- and local shielding disconnects the object (the losses w_{12} in this section are determined by a complete loss of effectiveness):

$$w_{12} = B^* t''.$$

The total losses for the second situation are

$$w_1 = w_{11} + w_{12} = (B_1^* - B_1)\,t' + B^* t'',$$

where $t' + t'' = T_{rep}$ and $\lim\limits_{t' \to 0} w_1 = B^* T_{rep}$.

When a UVM fails in a hierarchical system, control is not completely lost, but its effectiveness is reduced. As a result of a failure, let it take the value B_2. In this case, the loss w_2 from the loss of control optimality is

$$w_2 = (B_2^* - B_2)\,T_{rep}.$$

As a rule, it is considerably less than the loss suffered when a UVM in a centralized system goes out of order. The reduction in the loss in a hierarchical system is achieved by increasing its cost because of the presence of local regulators and the additional losses caused by their unreliability. In order to simplify the analysis, let us assume that $B_1^* = B_2^* = B^*$. The total expenses for centralized W_1 and hierarchical W_2 systems can then be evaluated in the following manner:

$$W_1 = (B^* - B_1)\,t'\lambda_0 + B^* t''\lambda_0 + C_0; \qquad (5\text{-}28)$$

$$W_2 = (B^* - B_2)\,T_{rep}\lambda_0 + \sum_{i=1}^{n} (b_i^* - b_i)\,\lambda_i t_i' + b_i^* \lambda_i t_i'' + C_0 + \sum_{i=1}^{n} C_i, \qquad (5\text{-}29)$$

where λ_0 = intensity of UVM failures; b_i^* = optimum efficiency value of the i-th circuit; b_i = nonoptimum efficiency value of the i-th circuit; λ_i = intensity of failure of the i-th local regulator; C_i = cost of the i-th local regulator.

In order to simplify the analysis, let us assume that

$$b_i = b_n;\ C_i = C_n;\ \lambda_i = \lambda_n,\ i = 1, 2, \ldots, n-1. \qquad (5\text{-}30)$$

For the i-th control circuit (allowing for the possibility that this circuit's regulator will go out of order), by analogy with a centralized control system (Figure 5-14), $t_i'' = t_{rep\ i} - t_i'$, where $t_{rep\ i}$ = time for repair of the i-th regulator; t_i' = time during which the i-th parameter does not ge beyond its allowable limits; t_i'' = time during which the i-th parameter is beyond its allowable limits.

In the first approximation, let us assume that $t_i' = t'$, $i = 1, 2, \ldots, n$.

Let us use the definitions $\mu_0 = 1/T_{rep}$; $\mu_n = 1/t_{rep\ i}$ = intensity of repair of the UVM and the local regulator, respectively; it is then the case that

$$\Delta_1 = (B^* - B_1)/B^*; \ \Delta_2 = (B^* - B_2)/B^*, \quad (5\text{-}31)$$

where Δ_1, Δ_2 = relative losses.

When (5-30) and (5-31) are taken into consideration, equations (5-28) and (5-29) can be written as follows:

$$W_1 = B^* [\Delta_1 \lambda_0 t' + \lambda_0 (1/\mu_0 - t')] + C_0; \quad (5\text{-}32)$$

$$W_2 = B^* [\Delta_2 \lambda_0/\mu_0 + \Delta_1 \lambda_n t' + \lambda_n (1/\mu_n - t')] + C_0 + nC_n. \quad (5\text{-}33)$$

Let us say that $\mu_0 t' = \eta_0$, $\mu_n t' = \eta_n$ and substitute them into (5-32) and (5-33):

$$W_1 = B^* \left\{\frac{\lambda_0}{\mu_0} [1 - \eta_0 (1 - \Delta_1)]\right\} + C_0; \quad (5\text{-}34)$$

$$W_2 = B^* \left\{\frac{\lambda_n}{\mu_n} [1 - \eta_n (1 - \Delta_1)] + \Delta_2 \frac{\lambda_0}{\mu_0}\right\} + C_0 + nC_n. \quad (5\text{-}35)$$

From (5-34) and (5-35) it follows that in the case of a hierarchical system, the total losses will be less than those for a centralized system only in the case where the following condition is fulfilled:

$$\lambda_n/\mu_n \ll \lambda_0/\mu_0. \quad (5\text{-}36)$$

This condition means that the local regulators' unreliability must be substantially less than the UVM's unreliability. In order that a hierarchical system be more efficient than a centralized one ($W_1 > W_2$), the local regulators' unreliability must be

$$\frac{\lambda_n}{\mu_n} < \frac{(\lambda_0/\mu_0)\,[1 - \eta_0 (1 - \Delta_1) - \Delta_2] - nC_n/B^*}{1 - \eta_n (1 - \Delta_1)}.$$

Let us find a graphic solution to equations (5-34) and (5-35), which determine the limit of the effectiveness of the use of one structure or the other. We will use the definition $(\lambda_0/\mu_0)/(\lambda_n/\mu_n) = \xi$, where ξ = relative unreliability of the computer in comparison to the reliability of the local regulators.

Let us transform (5-34) and (5-35) in the following manner:

$$\psi_1 = W_1/B^* = (\lambda_0/\mu_0) [1 - \eta_0 (1 - \Delta_1)] + C_0/B^*; \quad (5\text{-}37)$$

$$\psi_2 = W_2/B^* = (\lambda_0/\mu_0) \{(1/\xi) [1 - \eta_n (1 - \Delta_1) + \\ + \Delta_2]\} + C_0/B^* + nC_n/B^*. \quad (5\text{-}38)$$

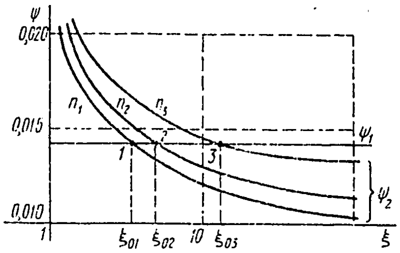


Figure 5-15. Dependence of effectiveness of centralized and hierarchical systems on the relative reliability of the computer and the local regulators.

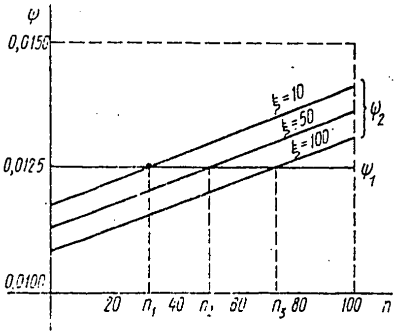


Figure 5-16. Toward determining the number of local regulators in a hierarchical system.

In Figure 5-15 we have plotted the graphs of the relationships $\psi_1 = \varphi_1(\xi)$ and $\psi_2 = \varphi_2(\xi)$, from which it follows that as ξ increases, the losses in a hierarchical system decrease in comparison with those in a centralized one (ξ_{01}, ξ_{02}, ξ_{03} = = points where the losses in both systems are identical):

$$\xi_{0i} = \frac{1 - \eta_{n}(1 - \Delta_1)}{1 - \eta_{0}(1 - \Delta_2) - n_i\mu_0 C_{ni}(\lambda_0 B^*)}. \quad (5\text{-}39)$$

From (5-39) it follows that the losses in a hierarchical system are linearly dependent on the number and cost of the local regulators. For a given level of relative computer unreliability ξ_i and local regulator cost C_n, the number of control circuits n can be chosen on the basis of the solution of equations (5-37) and (5-38). Figure 5-16 shows the results of this solution for three values of ξ and C_n = constant.

A hierarchical system will be more effective than a centralized one if the number of control circuits is $n_1 < n_{10}$ (for ξ_1), $n_2 < n_{20}$ (for ξ_2) and $n_3 < n_{30}$ (for ξ_3). The results obtained are correct only for systems evaluated solely according to the criterion of total expenses.

CHAPTER 6. MODERN UVM'S AND TRENDS IN THEIR DEVELOPMENT

6.1. Principles of the Construction of Modern UVM's and a Brief Description of Them

Despite the fact that the structures of various computers can differ from each other, it is possible to distinguish two methods for building both UVM's and general-purpose computers: fixed and modular.

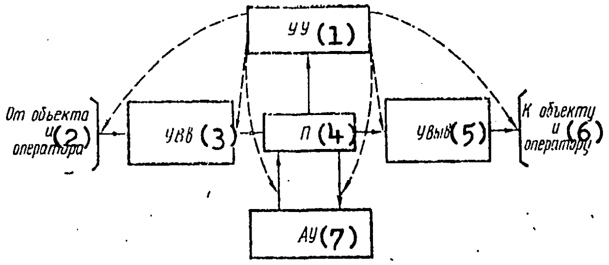


Figure 6-1. Diagram of UVM with fixed equipment structure.

Key:

1. Control unit UU
2. From controlled object and operator
3. Information input device UVv
4. Memory P
5. Information output device UVyv
6. To controlled object and operator
7. Arithmetic unit AU

When the first method is used, the structure of the computer's equipment is rigidly fixed and its enlargement beyond whatever was provided before when it was planned is impossible. The communication links between the computer's separate units (Figure 6-1, where UVv = information input device, P = memory, UU = = control unit, UVyv = information output device, AU = arithmetic unit) are also rigidly fixed. Construction of a computer with a rigidly fixed structure is regarded as the classical computer construction principle. This principle has been realized in both first- and second-generation general-purpose and control computers. The basic advantage of computer construction with a fixed equipment structure is the relative simplicity of organizing information processing. This makes it possible to have a comparatively simple computer control unit, which in the final account leads to simplification of the equipment realization of the computer as a whole. The disadvantage of the fixed structure is the impossibility of expanding beyond a certain limit the volume and nature of the problems that can be solved.

As a rule, in the case of evolution of solvable problems toward greater complexity, systems built on the basis of computers with a fixed equipment structure require complete replacement of the computing equipment.

The second method of constructing computers is characterized by the absence of rigid fixation of the structure of the hardware from which the computer is assembled. Only the equipment in the computing unit -- the processor, which is the heart of the computer -- is fixed on a certain level. In order to produce computers with different productivity rates, a whole series (family) of processors with different operating speeds are being developed. The peripheral equipment, including information input and output devices, external memories, and units for communicating with operators, are not rigidly fixed and can be changed within wide limits. A characteristic feature of such a structure is the use of special devices for linking the separate units (parts) into a unified system. These coupling devices are called input-output channels. Such a channel is a control unit that insures commutation and communication of the peripheral equipment with the computer's nucleus. Two types of channels are used: selector (SK) and multiplex (MK). Channels are categorized as selector or multiplex depending on their ability to serve several peripheral units simultaneously.

A selector channel is used to transmit or receive information at a given moment to only one peripheral device or from one of the peripheral devices connected to it. Until its work with the one peripheral device has been completed, the others must stand idle or perform auxiliary operations that are not connected with the information trunk, which is occupied by the operating device at that time.

A multiplex channel is intended for alternate use by several peripheral devices operating in parallel. Information transmission between the peripheral devices and the memory is carried out with the help of comparatively brief communication periods during which a small amount of information is passed.

The communication periods between the different peripheral devices and the memory alternate with each other.

As a rule, a computer is equipped with both selector and multiplex communication channels. Selector channels provide the highest information transmission rates, so the relatively high-speed peripheral devices (magnetic drums, disks and tapes) are connected to them.

Thus, this principle of computer construction assumes the use of a processor with a certain productivity rate and a set of peripheral devices and an operational memory that correspond to this rate. Such computer organization is typical of modern, third-generation computers. The method of building computers from separate units is called modular construction.

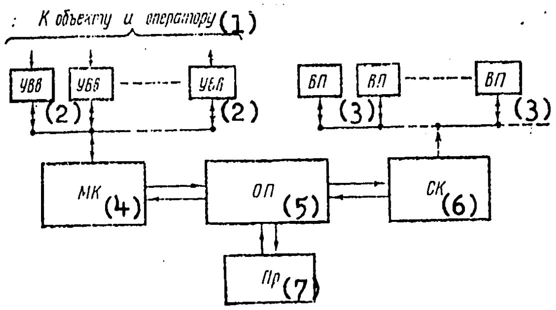


Figure 6-2. Diagram of UVM built according to the modular principle.

Key:

1. To controlled object and operator
2. Input-output devices UVV
3. External memory VP
4. Multiplex channel MK
5. Operational memory OP
6. Selector channel SK
7. Processor Pr

When the modular construction method is used, the separate functional units are realized in the form of independent modules that are combined in the required quantities and types to form the computer. In essence, when the modular principle is used we should speak not of a computer, but of a computing system. Figure 6-2 is a structural diagram of a computer built on the modular principle: UVV = input-output devices; MK, SK = multiplex and selector channels, respectively; OP = operational memory; Pr = processor; VP = external memory.

The use of the modular construction principle yields particularly tangible results in a UVM, since control systems must be open to constant evolution. This is inevitable as experience in operating control systems and knowledge of the process or object being controlled are accumulated. Hardware modularization makes it possible to insure the equipment adaptability of a control system under changing operating conditions.

A control system's capabilities can be expanded or it can be modernized either by simply adding equipment to the controlling part of the system, or changing the composition of the modular units of which it is comprised, or changing to new modular units from the same family of modular computer hardware that are more productive, have a larger memory capacity, and so on.

The realization of the modular principle of computer construction is based on the standardization of both internal and external communications. Communication standardization is insured by fulfilling the conditions of physical and mathematical compatibility of the modular units in the system. Physical compatibility presumes the organization of information transmission between the system's units by signals having strictly defined parameters. Mathematical compatibility is taken to mean the information and program compatibility of computers composed of modular system units with different productivity rates. Information compatibility presumes unified encoding of information for all the units making up the modular system. As a result, some day encoded data can be processed by a computer composed of any units in a given family of modular hardware. Program compatibility presumes the possibility of using programs written for one computer in all the other computers belonging to the same family.

6.2. Soviet Modular Hardware Systems

In our country, the idea of modularization in computer construction was most fully embodied in the development of the two series of computers called the ASVT (Modular System of Computer Technology) and the YeS EVM (Unified System of Electronic Computers). The computers in these series not only have different technical characteristics, but are also used for different purposes.

The ASVT is primarily oriented for use in control systems, including those for the control of technological processes and objects.

The YeS EVM is intended for use in computer centers, for the solution of both a variegated series of computational problems and data processing problems. The computers in this system can also be used in automated organizational and administrative control systems on different levels and for different purposes.

YeS EVM system computers do not have facilities for direct communication with a controlled object.

The following installations are created on the basis of ASVT computers: 1) integrated systems for the automatic monitoring and control of technological processes and separate technical objects in different branches of industry; 2) automated control systems for shops, industrial enterprises, electric power stations, and power systems; 3) automated queuing systems; 4) enterprise computer centers, and so on.

ASVT UVM's are realized in the form of two systems: ASVT-D and ASVT-M. The ASVT-D system is composed c‿ discrete semiconducting elements, but has the same structure as third-generation computers. The ASVT-M system is realized from micromodular elements and is a further development and improvement of the ASVT-D computers.

Let us examine the operating principles and structure of ASVT-D series computers, since the first use of the modular construction principle in Soviet practice was in this system.

An ASVT-D computer consists of units that perform the following operations: 1) central information control and processing (processors); 2) information storage (operational and external memories); 3) communication with the controlled object; 4) communication with the operating personnel; 5) input from information carriers (punched cards and punched tape) and output on them; 6) intrasystem (intracomputer) communication; 7) output on communication links outside the system.

Information compatibility is realized by standardizing the unit of information transmitted among the elements. The unit normally adopted for use is the byte, which equals eight binary digits (bits). A byte can be used to encode 256 messages (pieces of information) or two 10-digit numbers. By providing a byte with a single check bit, it is possible to monitor the information transmission channels for parity. Two bytes form a half-word. Two half-words form a word. It is also possible to use double-length words (64 bits). The smallest addressable unit of information in an ASVT-D is a byte. The address of each word in the memory is determined by the address of the byte to the extreme left. The length of the word used to perform an operation is given in the command that is being executed. The total addressable information capacity in an ASVT-D system is about 17 Mbytes, which makes it possible to solve a very broad circle of both scientific and technical and production problems.

Physical compatibility is realized by regulating the signals' parameters and standardizing the intrasystem communications. There are three types of such standardized communications in the ASVT system. Each of the types of communication is called a rank. Depending on the number and purpose of the communication links they contain, ranks are divided into linkages. The direction of the linkage is indicated by a special separation of them into central and peripheral sides. Figure 6-3 shows the structure of the standardized communications in an ASVT-D computer.

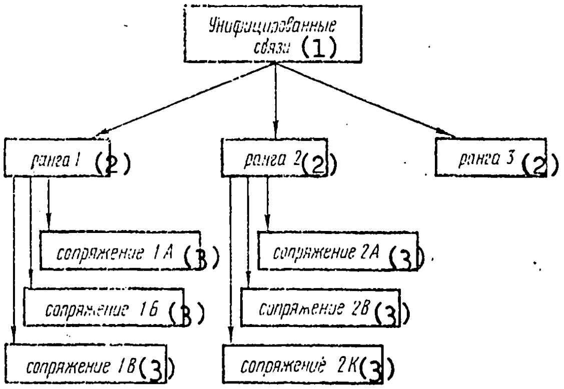


Figure 6-3. Structure of standardized ASVT communications.

Key:

1. Standardized communications
2. Rank .
3. Linkage ..

Rank 1 communications are intended to connect units with an intensive exchange of information (the processor, operational memories, memory-shielding devices). The maximum rank communication distance is 1-10 m. Information is transmitted in parallel. Information from different sources is passed into a common trunk. The moments for the transmission of information into the trunk from the individual sources are determined by a special control circuit that is part of Communications Rank 1. Information reception from the trunk by separate units in the computer is accomplished with the help of the associative indicator carried by each informational word passed into the trunk.

Communications Rank 2 is used to unite devices with a less intensive type of information exchange. These links connect the processor and the input-output units. The method of transmission over Rank 2 is byte-by-byte, in parallel. Each byte is provided with additional service signals that define its purpose. Communications Rank 2 is subdivided into Linkages 2a, 2C and 2K. As is the case with Communications Rank 1, Linkage 2A has a trunk structure. Linkage 2C is intended for the organization of radial connection of the individual elements. At each moment, one of the external units can be connected to the communication system. Linkage 2K is used in series ASVT-M computers (the M-6000 complex). The Communications Rank 2 system is organized on an asynchronous principle and provides information transmission in response to an inquiry signal.

Communications Rank 3 is used to transmit information over long distances (up to 15 km). The method of transmission over the links in Rank 3 is sequential. The unit of transmitted information is the binary digit (or bit). The rate of transmission is 5,000 bits/sec. Since the transmitted information is monitored, its effective transmission rate (without allowing for the transmission of the monitoring and service information) is 2,000 bits/sec. The transmission channel is a special three-wire cable.

Two types of processors -- specialized and general purpose -- are used for centralized information control and processing in an ASVT-D computer.

A specialized processor is one that insures the processing of comparatively small masses of information (solving simple scientific and technical problems, primary information processing, and so forth). It can also be used in complicated computer complexes. The characteristic feature of specialized processors is a combination of information processing functions with functions for exchanging information among the input-output devices. In order to accomplish these functions, the processor has outputs into Linkages 1A, 1B and 2A.

A general-purpose processor is used to solve complicated control problems (such as optimization, technical-economic and operations-production planning, material and technical supply control). In general-purpose processors, communication with operational memories is carried out over Linkages 1A and 1B, while communication with the information input-output units is accomplished through multiplex channels that are part of Linkage 2A. Each multiplex channel provides communication with up to 16 input-output devices.

The ASVT-D system features an M-1000 specialized processor and M-2000 and M-3000 general-purpose processors. Possible circuits for connecting specialized and general-purpose processors to internal (operational) memory units and input-output units are shown in Figure 6-4a, b and c, where 1A, 1B and 2A = linkages of Ranks 1 and 2; US-1M = Communications Rank 1 unit;

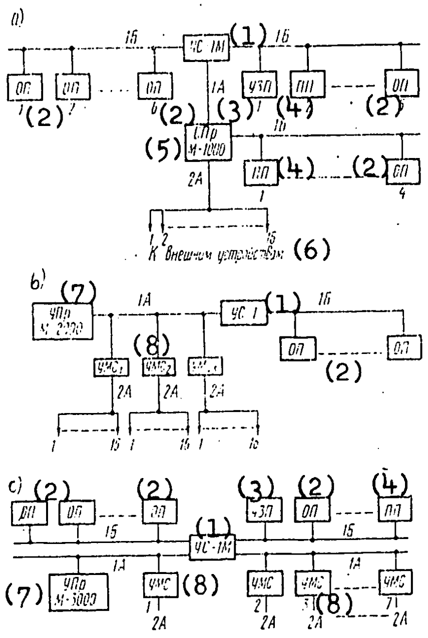


Figure 6-4. Structure of M-1000 (a), M-2000 (b) and M-3000 (c) complexes.

Key:

1. Communications Rank 1 unit US-1M, US-1
2. Operational memory OP
3. Memory-protection unit UZP
4. Permanent memory PP
5. M-1000 specialized processor SPr
6. To peripheral gear
7. M-2000 and M-3000 general-purpose processors UPr
8. Multiplex communications unit UMS

SPr = specialized processor; OP, PP = operational and permanent memories, respectively; UMS = multiplex communications unit; US-1 = Communications Rank 1 unit; UPr = general-purpose processor; UZP = memory-protection unit.

The ASVT-D system provides for program compatibility of all general-purpose processors. The specialized M-1000 processor is program-incompatible with other processor models.

For the practical realization of program compatibility of different models of processors, it is necessary to insure their functioning with that set of peripheral gear specified in a program.

The general software contains programs for realizing individual utilization modes, package processing, and multiprogram and multiprocessor modes. The first stage of the automated programming system, as realized in the ASVT-D, specifies: 1) an upper-level external algorithmic language (ALGEK) that is a subset of ALGOL-60 and that has been expanded through the introduction of additional means that make it possible to describe documents, masses of documents, and the processes for handling the information in them quite easily; 2) a middle-level computer-oriented language (ALMO); 3) a library of standard subprograms for solving the most typical computation and control programs.

The ASVT's operational system realizes the following programs for controlling the computation process: 1) interruption; 2) memory allocation during a multiprogram operating mode; 3) memory protection; 4) operating time assignment; 5) control of the input-output process, and others.

As a rule, all the operating system's programs are stored in the permanent memory. For the future, it has been proposed that the software be expanded substantially by using translators with other external algorithmic languages and a disk operating system (an operating system on magnetic disks). The latter should be used primarily in an ASVT-M system.

The ASVT-M system includes: a) M-6000, M-7000, M-400, M-4000, M-4030, and M-5000 UVM's (complexes); b) M-6010 and M-40 microprogram automatic control units.

According to the nature of the input information that is used and the output information that is formulated, ASVT-M series UVM's can be divided into two classes:

1. UVM's for processing (in real time) input and formulating output information in the form of electrical signals of a different type (M-6000, M-7000, M-6010, M-400, and M-40). The UVM's in this class are primarily intended for use in control systems that are on the level of complex technological processes and objects, as well as production facilities and shops. In a minimal configuration they can collect, process and present information in digital form through special input-output devices (keypunch machines, printers and teletypes).

2. UVM's for processing (in real time) input and formulating output information in digital form. In connection with this, the input information is entered in the UVM either by an operator through input-output devices or through a lower-level UVM. Information output is performed analogously. Higher-level control systems are built on the basis of computers of this type. Higher-level UVM's include the M-4000 (the first version of the ASVT-M system), M-4030 and M-5000 complexes.

A large set of commands, a modern software system, and the presence of means for processing alphanumeric information make it possible to use higher-level ASVT-M computers of different types as a basis for forming automatic production control systems, and they can also be used to process data in enterprise computation centers.

Figure 6-5 depicts the logic structure for a type M-4030 UVM, where AU = arithmetic unit; SOP = supernormal memory; BMK = = microcommand block; IPU = technical control panel; MK, SK = = multiplex and selector channels, respectively; PK, PL =

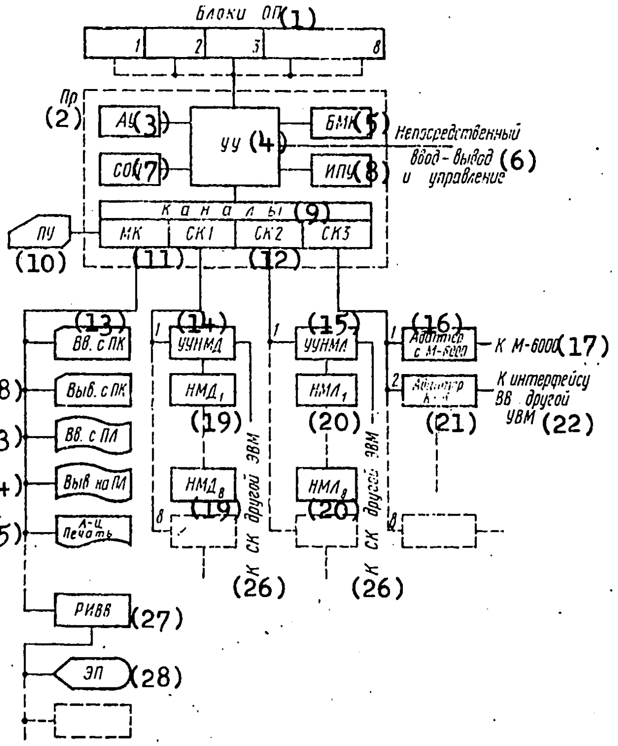


Figure 6-5. Structure of ASVT-M M-4030 complex.

Key:

1. Operational memory blocks
2. Processor Pr
3. Arithmetic unit AU
4. Control unit UU
5. Microcommand block BMK
6. Direct input-output and control
7. Supernormal memory SOP
8. Technical control panel IPU
9. Channels
10. Control panel PU
11. Multiplex channel MK
12. Selector channels SK.
13. Input from punched cards PK
14. Magnetic disk storage control unit UUNMD
15. Magnetic tape storage control unit UUNML
16. Adapter to M-6000
17. To M-6000
18. Output on punched cards
19. Magnetic disk storage NMD.
20. Magnetic tape storage NML.
21. Channel-channel adapter
22. To input interface of another UVM
23. Input from punched tape PL
24. Output on punched tape
25. Alphanumeric printer
26. To SK of another computer
27. Input-output interface expander RIVV
28. Display panel EP

= punched cards and punched tape, respectively; UUNMD, UUNML = = magnetic disk (NMD) and magnetic tape (NML) storage control units, respectively; K-K = channel-channel; RIVV = input-output interface expander; EP = display panel; PU = control panel; A-Ts = alphanumeric printer.

From the plan that has been presented it is obvious that the structure of an M-4030 UVM completely matches that of a general-purpose computer. The only special device is the unit for communicating with ASVT-M M-6000 complexes. The basic specifications of the M-4030 control complex are presented in greater detail in Appendix 6.

ASVT-M series UVM's of the M-6000, M-7000, M-6010, M-400, and M-40 type differ substantially from their predecessors -- ASVT-D and ASVT-M (M-4000, M-4030) series computers -- in both logic structure and technical characteristics. These computers belong to the small UVM class (mini-UVM's).

6.3. Mini-UVM's

The problem of the effective utilization of computers in different automated and automatic control systems can be solved in two ways. The first way presumes the use of high-productivity computers operating in a time-sharing mode and servicing a considerable number of users. The second way consists of using a large number of small computers (mini-UVM's). In connection with this, it is necessary that the volume of each problem to be solved does not require the use of large computers. In many cases the use of small computers -- particularly for solving production process control problems -- is preferable, since they are easier to acquire and maintain, while the one-time expenses for organizing control systems based on them are significantly less than those for large computers.

In a number of cases, a large number of calculations are simply not required for the solution of control problems. Such problems include the logic problems that are very often encountered in control systems, the sorting and primary processing of information, and others. The use of small computers or computational systems composed of several small computers can also prove to be economically advantageous in comparison with the use of medium- or high-productivity computers, since a computer's productivity is approximately proportional to the square of its cost.

Despite the fact that we cannot delineate clear boundaries between large, medium and small computers, it can be assumed that the following parameters are typical for minicomputers: 1) word length (word format) -- 8-16 (18) bits; 2) a binary number system; 3) operations on numbers with a fixed decimal point; 4) an operational memory volume of from (2-4)K to (32-64)K words; 5) a wide assortment of connectable input-output devices; 6) time for performance of basic arithmetic operations -- 1.5-4 μsec or less; 7) low cost and high reliability for the minimum computer configuration; 8) small size.

The cost of small computers is reduced and their reliability is increased by reducing the word length of the information being processed, simplifying the command system, and using a processor to control the information input-output units.

For all practical purposes, the first control minicomputer in the world was the Soviet UMI-NKh type UVM, which was developed in the mid-1960's. Minicomputers received their greatest development in connection with the changeover to a microelectronic element base. In this case, the basic requirements for computers of this type, as outlined above, can be satisfied most fully. Since minicomputers have extremely good prospects as control computers, let us discuss in more detail the structure, construction features and technical characteristics of the most typical Soviet control minicomputers. Computers of the "Elektronika K-200" and M-6000 (ASVT-M system) types are characteristic representatives of Soviet control minicomputers.

The Elektronika K-200 Minicomputer. This control computer is built on a microelectronic base and is oriented for use in complex integrated, optimal and logic-program control systems, as well as enterprise computation centers (for preliminary information processing). Figure 6-6 is a diagram of this computer, with PU = control panel; UK = monitoring unit; AU = arithmetic unit; SOP = high-speed memory; TsUU = central control unit; SchK = command counter; RgK = command register; BPRP = program interrupt unit; OP, PP = operational and permanent memories, respectively.

Special features of this computer's structure are the presence of a special, built-in input-output control block (BUVV), which provides multiplex and selector modes for exchanging information with peripheral gear, and the use of the permanent memory unit to store programs for solving functional control problems. The presence of a permanent memory for storing functional control problem solution programs makes it possible to use the fast operating speed of the units in the computer's operating part more efficiently and thereby makes the realization of the control programs in real time easier. Thus, a UVM of the Elektronika K-200 type can be used in the programmed automated machine mode. The latter is most often realized when the computer is used to control production processes.

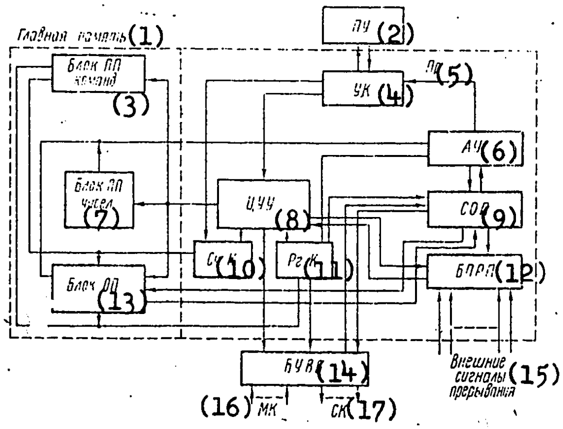


Figure 6-6. Structural diagram of Elektronika K-200 UVM.

Key:

1. Main memory
2. Control panel PU
3. Permanent memory PP command block
4. Monitoring unit UK
5. Processor Pr
6. Arithmetic unit AU
7. Permanent memory PP number block
8. Central control unit TsUU
9. High-speed memory SOP
10. Command counter SchK
11. Command register RgK
12. Program interrupt unit BPRP
13. Operational memory OP block
14. Input-output control block BUVV
15. External interrupt signals
16. Multiplex channel MK
17. Selector channel SK

As in many third-generation computers, in the Elektronika K-200 it is possible to expand the computer's capabilities by enlarging the capacity of the operational memory (from 1K to 16K with 512-word blocks) and the permanent memory (from 3K to 16K with 2K blocks). Each block in the operational memory has inputs for internal and external exchanges of information.

Special design features of this computer are the use of hybrid microcircuits with element-by-element duplication and the division of the computer's control panel into two parts: 1) an internal adjusting and monitoring panel; 2) a working panel, into which lead the minimum number of control elements that will insure the computer's operation in the normal operating mode. Access to the adjusting and monitoring panel is blocked by a special key, which makes it possible to prevent random interference in the computer equipment's operation by nonspecialists when (for example) the computer is operating under shop conditions.

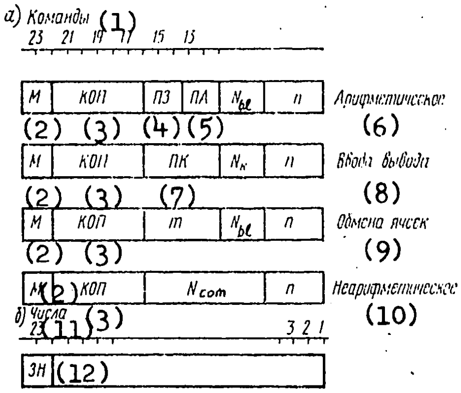


Figure 6-7. Structure of representation of command (a) and numerical (b) information in Elektronika K-200 computer.

Key:
1. Commands
2. Modification indicator M
3. Operation code KOP
4. Transferrence indicator PZ
5. Address indicator PA
6. Arithmetic
7. Channel indicator PK
8. Input-output
9. Cell exchange
10. Nonarithmetic
11. Numbers
12. Character cell ZN

A binary number system is used in the Elektronika K-200. The number representation method is with a fixed decimal. The word length is 23 bit positions, with 1 additional position in the number memory (for automatic parity checking). The structure of command and numerical information representation is shown in Figures 6-7a and b, respectively. The performance of 32 operations is provided for: 16 arithmetic, 4 input-output, 2 exchange, and 10 nonarithmetic. The following designations are used in Figure 6-7: M = modification indication; KO = operation code; PZ = = indicator of transferrence into high-speed memory's register; PA = address indicator; PK = channel indicator; m, n = = high-speed memory's address register; N_{bl} = address in block; N_k = channel address; N_{com} = complete address; ZN = = character cell.

The Elektronika K-200 UVM is equipped with external software that uses a general-purpose M-220 computer as a programming computer. The external software includes a modeling program of the Elektronika K-200 computer's system of commands, which makes it possible to adjust the programs outside the UVM. The results of the adjustment -- altered program cards -- are entered in the computer's permanent memory. Besides this, in the computer there are provisions for the use of an autocoder and a library of standard subprograms, as well as a series of other service and utility programs.

An Elektronika K-200 UVM is equipped with an extensive set of peripheral gear, which makes it possible to form different standard complexes using this gear as a base. The equipment composition of a standard complex is determined by the conditions of its utilization in the system.

In the information mode, the standard complex contains: a) the UVM itself (processor, internal memory); b) up to 8-16 ATsPU's [alphanumeric printer]; c) digital input-output (printer, photodiode reader, tape punch); d) an operational memory of up to 8,000-16,000 words and a permanent memory of up to 4,000 words.

In the adviser mode, the standard complex contains: a) the UVM itself (processor, internal memory); b) up to 2-4 ATsPU's; c) an analog-to-digital "current-to-code" converter; d) a unit for visual depiction of information on a cathode-ray tube; d) an operational memory of up to 2,000 words and a permanent memory of up to 6,000 words.

In the control mode, the standard complex is made up of: a) the UVM itself (processor, internal memory); b) 1 or 2 ATsPU's; c) analog-to-digital and digital-to-analog converters; d) a permanent memory of up to 8,000-16,000 words.

Other complexes can also be formed on the basis of the standard ones.

The Type M-6000 ASVT-M Complex. The set of aggregate modules in the ASVT-M M-6000 is used to make up, according to plans, autonomous information and control computer complexes that operate in real time and are characterized by comparatively simple information processing algorithms.

As with other minicomputers, the M-6000 uses the binary form of number representation, with a fixed decimal. The basic command and information representation form is the half-word (16 bits); it is also possible to operate with whole words (32 bits). The structure of command and numerical information presentation is shown in Figure 6-8.

The processor executes 72 commands. All commands are divided into three groups: address (AK), register (RK) and input-output (VV). There are also provisions for operation according to special subprograms with numbers having a floating decimal, as well as the processing of symbol-type (alphanumeric) information.

One special feature of the M-6000 minicomputer is the possibility of operating with accelerated performance of multiplication and division operations. For this purpose, a special circuit called an arithmetic expander (RA) is introduced into the processor. Thus, if the execution of multiplication and division operations takes 190 and 480 μsec, respectively, without the expander, with it this time is reduced to 50 μsec for multiplication and 60 μsec for division.

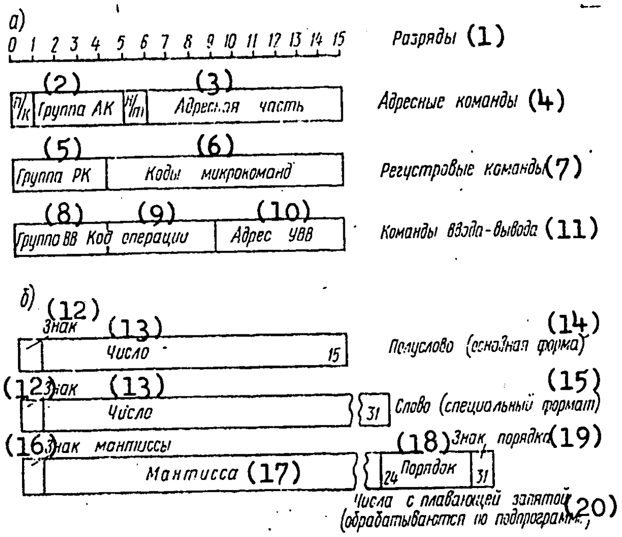


Figure 6-8. Structure of presentation of command (a) and numerical (b) information in ASVT-M M-6000 complex.

Key:

1. Bit positions	12. Sign
2. AK group	13. Number
3. Address part	14. Half-word (basic format)
4. Address commands AK	15. Word (special format)
5. RK group	16. Sign of mantissa
6. Microcommand codes	17. Mantissa
7. Register commands RK	18. Exponent
8. VV group	19. Sign of exponent
9. Operation code	20. Numbers with floating decimals (processed with subprograms)
10. Address of input-output unit	
11. Input-output commands VV	

Analogous expanders are also used in the information input-output system (input-output expanders = RVV). These expanders make it possible to increase the number of channels, with the help of which a significantly greater number of peripheral devices (input-output units = UVV) can be connected to the system. Thus, if without the expander it is possible to connect 8 peripheral devices, when 1, 2, or 3 expanders are used, this number increases to 22, 38, or 54, respectively (Figure 6-9).

All connections of peripheral gear with the processor are made with the help of a standardized 2K linkage, which is realized

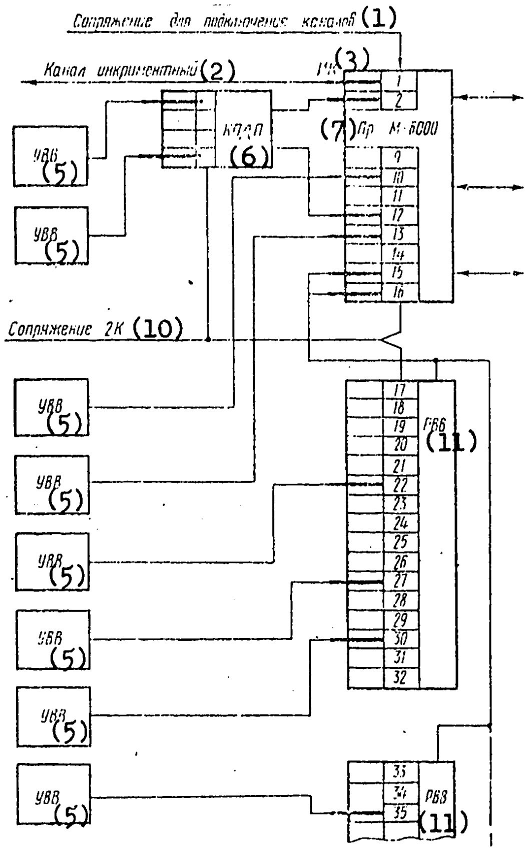


Figure 6-9. Block diagram of connection of input-output units to M-6000 computer's processor.

Key:

1. Linkage for connecting channels	6. Direct memory access channel KPDP
2. Increment channel	7. Processor Pr
3. Interface cards IK	8. Operational memory OP
4. Arithmetic expander RA	9. Permanent memory PP
5. Input-output units UVV	10. 2K linkage
	11. Input-output expander RVV

in the form of a control circuit that is connected to the processor or the input-output expander. One or two plates called interface cards (IK), which also connect the processor to the input-output devices, are provided for each of the channels. Special matching units are provided in order to match the signal levels during transmission from the external unit to the processor and vice versa.

In addition to transmitting information through the processor in the M-6000, there is also a channel for direct access to the memory (KPDP) that makes it possible to enter and transfer information directly into the operational memory (OP) without loading the processor (and to receive and transfer information from the OP into one of the peripheral units). However, control information is transmitted only through the processor. This type of input-output unit connection is called connection to the program channel. The KPDP's data transmission rate is up to 400,000 words/sec.

The general structure of the peripheral gear for the ASVT-M M-6000 complex is shown in Figure 6-10.

On the basis of the processor and the peripheral gear, it is possible to put together both specialized and 24 standard and basic complexes.

The composition of the equipment, the logic structure of the connection of the units, and the configuration of the first five complexes are presented in Appendix 4.

In order to increase the productivity or vitality of the system, a multiprocessor operating mode is provided for in the M-6000.

Special matching units can be included in the M-6000 complex in order to connect it to other types of processors in the ASVT-M system and to computers in the YeS EVM system.

The software system provides for the following operating modes: program preparation, program debugging, solution on an unnatural time scale, multiprogram operation on a real time scale, preventive maintenance and searching for defects.

The programming automation system provides for the use of the ASVT-M-6000 mnemonic code and the FORTRAN and ALGOL-60 (or, to be more precise, a dialect of ALGOL-60) external languages. The appropriate translators are provided for the purpose of translating programs from these languages. For the future, it has been proposed that the system's capabilities be expanded by including other languages (BEYSIK, in particular) in it in order tc insure operation in a dialog mode with the operator. In addition to the external languages, the ASVT-M (M-6000 complex) programming automation system includes a library of standard problem-oriented programs.

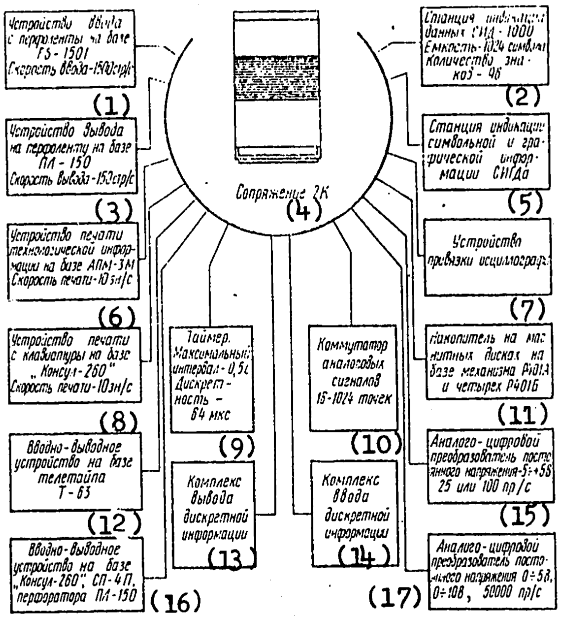

Устройство ввода с перфоленты на базе FS-1501 Скорость ввода-1500стр/с
(1)
Станция индикации данных СИД-1000 Емкость-1024 символа Количество знаков-96
(2)
Устройство вывода на перфоленту на базе ПЛ-150 Скорость вывода-150стр/с
(3)
Сопряжение 2К
(4)
Станция индикации символьной и графической информации СИГДа
(5)
Устройство печати технологической информации на базе АПМ-3М Скорость печати-10зн/с
(6)
Устройство привязки осциллографа
(7)
Устройство печати с клавиатуры на базе „Консул-260" Скорость печати-10зн/с
(8)
Таймер. Максимальный интервал-0,5с Дискретность-64 мкс
(9)
Коммутатор аналоговых сигналов 16-1024 точек
(10)
Накопитель на магнитных дисках на базе механизма Р401А и четырех Р401Б
(11)
Вводно-выводное устройство на базе телетайпа Т-63
(12)
Комплекс вывода дискретной информации
(13)
Комплекс ввода дискретной информации
(14)
Аналого-цифровой преобразователь постоянного напряжения-5÷+58 25 или 100 пр/с
(15)
Вводно-выводное устройство на базе „Консул-260", СП-4П, перфоратора ПЛ-150
(16)
(17)
Аналого-цифровой преобразователь постоянного напряжения 0÷58, 0÷108, 50000 пр/с

Figure 6-10. Structure of peripheral gear for ASVT-M M-6000 complex.

Key:

1. Unit for input from punched tape based on FS-1501 (rate of input = = 1,500 lines/sec)
2. SID-1000 data display station (capacity = 1,024 symbols; number of symbols = 96)
3. Unit for output on punched tape based on PL-150 (rate of output = 150 lines/sec)
4. 2K linkage
5. SIGDa symbol and graphic information display station
6. Unit for printing technological information based on APM-3M (printing rate = = 10 symbols/sec)
7. Oscillograph tie-in unit
8. Unit for printing from keyboard, based on Konsul-260 (rate of printing = 10 symbols/sec)
9. Timer (maximum interval = = 0.5 sec; discreteness = = 64 μsec
10. Analog signal commutator (16-1,024 points)
11. Magnetic disk storage unit based on R401A unit and four R401B units
12. Input-output unit based on T-63 teletype
13. Discrete information output complex
14. Discrete information input complex
15. Analog-to-digital direct-current converter (-5 to +58; 24 or 100 conversions/ /sec)
16. Input-output unit based on Konsul-260, SP-4P and PL-150 perforator
17. Analog-to-digital direct-current converter (0-58, 0-108; 50,000 conversions/sec)

The ASVT-M operating system (the M-6000 complex) provides for operation in the multiprogram and operator-dialog-with-computer modes. Besides this, the software system provides for the execution of a number of other operations (accounting for current time, carrying out assignments for starting and stopping programs at a given time and at given intervals, automatic output and output by request of needed information on the operator-technician's panel, turning on diagnostic programs and test problems, and so forth).

The M-7000 system is a further development of the ASVT-M system. Its basic technical characteristics and the structure of its three basic complexes are given in Appendix 5.

6.4. Basic Trends in the Development of UVM's

We make this evaluation of the basic trends in the development of UVM's from the viewpoint of the changes in their architecture, equipment realization, software, and areas of application.

In general and on the whole, the structure or architecture of modern UVM's corresponds to that of general-purpose computers. As is the case with computers, the principle of modularization is used with UVM's; that is, the use of a central processing unit with a freely formulated periphery. As has already been mentioned, the use of a nonrigid structure makes it substantially simpler to solve problems relating to a control system's evolution toward greater complexity and improvement of the algorithms.

In order to satisfy the productivity requirements, a program-compatible series of processors are being provided that are distinguished by their high operating speed and make it possible to work with different numbers of external units and operational memory blocks.

In connection with the appearance of organization and administrative control problems related to the processing of large masses of data, efforts have been made to expand the concept of the control computer into the class of data-processing computers. This attempt is manifested most clearly in the creation of complexes of modular hardware capable of solving control problems on different levels, beginning with the control of machinery and ending with the solution of organizational and administrative control problems.

Along with the development of sufficiently powerful computers, we should expect constant improvement and increases in the output of small control computers capable of solving the problem of creating local control systems as quickly as possible and with comparatively small expenditures. In view of the fact that a very large proportion of the assignments in control systems involve the collection, simplest processing, and presentation of information, small UVM's should also be constructed with a nonrigidly fixed peripheral gear structure. In contrast to high-productivity computers, small UVM's are not built in the form of a series of program-compatible processors with different productivity levels. As far as the methods used to connect and operate peripheral equipment are concerned, their structure fully reflects the structural design of third-generation modular systems. We should expect that this trend will also be preserved in the future.

It is possible to point out yet another characteristic feature of the structure of small UVM's. All of these computers are built so they can operate in the multiprocessor mode. Two goals are pursued in connection with this: 1) increasing (where necessary) the productivity of such a system; 2) increasing the operational reliability of the system's control section through various forms of redundancy.

The latter problem is solved particularly well by the use of small computers, which make it possible to build highly reliable control systems at a suitable cost.

The equipment realization of UVM's reflects the attempt to improve the operational reliability of computers and reduce material consumption and size while retaining their relatively fast operating speed. These goals have been achieved by a changeover to the use of integrated elements and large-scale integrated circuits (BIS). In connection with this, because of the improvement in technology there was not only an improvement in the reliability of a computer as a whole, but also a reduction in its cost, which is a factor of no little importance for the technical and economic substantiation for the use of any control system with a UVM.

In the mid-1970's, the developers of control and information-processing systems had at their disposal a new class of hardware characterized by the use of BIS's with a degree of integration of several thousand elements in a single monolithic circuit. BIS-based units for processing information received the name of microprocessors and microcomputers.

The Microprocessor. A microprocessor is a program- or microprogram-controlled processor realized with one or several BIS's. The classes of microprocessors that are distinguished are multipurpose and specialized.

In multipurpose microprocessors, the information processing algorithm is completely determined by an external program stored in the permanent or semipermanent memory blocks.

A specialized microprocessor is used to solve problems of a single type, so its operating program is realized by the equipment method. Such a decision makes it possible to build extremely high-speed information processing devices.

The basic purpose of microprocessors is to make up microcomputers and special circuits for local information processing in different units and systems such as digital regulators, measuring instruments and so on.

Structurally, microprocessors consist of: 1) an independent unit fitted with a connector, with the help of which it can be connected to functional units with different purposes; 2) modular elements that can be used to build processors for microcomputers with different capacities.

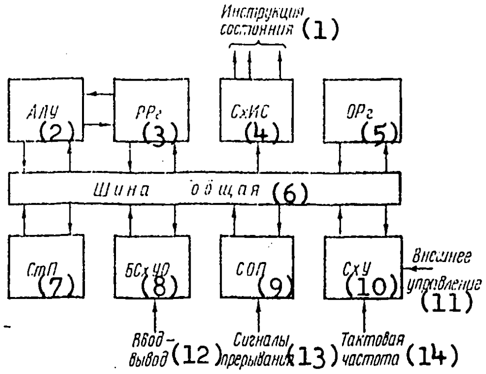


Figure 6-11. Generalized block diagram of a microprocessor.

Key:
1. Condition instruction
2. Arithmetic-logic unit ALU
3. Working registers RRg
4. Condition indication circuit SkhIS
5. Common registers ORg
6. Common line
7. Stack (magazine) memory StP
8. Exchange control buffer BSkhUO
9. Interruption processing circuit SOP
10. Control circuit SkhU
11. External control
12. Input-output
13. Interrupt signals
14. Clock frequency

Figure 6-11 is a generalized block diagram of a microprocessor, where ALU = arithmetic-logic unit; ORg = common registers;

StP = stack (magazine) memory; BSkhUO = exchange control buffer; SOP = interruption processing circuit; SkhU = control circuit; SkhIS = condition indication circuit; RRg = working registers.

One special feature of the structure of a microprocessor is the extensive use of registers for both buffer storage and organizing indirect addressing when working with a large-volume operational memory; another is the presence of a register magazine (stack) memory for storing subprograms, tables and other information.

Microprocessors are built to process words that are 4, 8, 12 and 16 bits long. The time required to perform short operations is 1-10 μsec.

Table A7-1 gives the basic specifications of several types of foreign microprocessors [65]. We should mention here the trend toward greater complexity in the functions assigned to microprocessors and the convergence of their specifications with those of minicomputer processors.

The Microcomputer. A microcomputer is a program-controlled, structurally complete computer that is composed of a microprocessor set of BIS's, operational (OP) and permanent (PP) memory blocks, and control units.

The term "micro" reflects the fact that at the present stage of hardware development, computers of this class are as light and small, consume as little power, and cost as little as possible. It should be mentioned, however, that the productivity of microcomputers is also low.

The characteristic features of microcomputers are: 1) realization of all functional units with BIS's; 2) maximally simple organization of information processing, using a single-program mode and specialized programs entered in the permanent memory; 3) a small number of information presentation units (up to 16 bits).

A generalized block diagram of a microcomputer is presented in Figure 6-12, where MPr = microprocessor; ZG = master clock; BSOD = data exchange buffer; BSPA = address transmission buffer; OP, PP = operational and permanent memories, respectively; SkhVV = information input-output circuit.

In microcomputers, the PP is used to store functional problem solution programs, while the OP is used to store the original data, intermediate calculations, and results. Both the OP and the PP are realized on the basis of BIS's. The use of a BIS-based PP to store programs determines the effectiveness of microcomputers in large production runs; that is, in systems for mass use.

Structurally, a microcomputer has the following form: 1) a system assembly (plate) fitted with a connector so it can be connected to elements of the common circuit of which the microcomputer is a component part; 2) an autonomous unit equipped with a monitoring and control panel.

The small dimensions, light weight, and low power consumption and cost mean that microcomputers can be widely used in control systems that previously used a rigidly fixed ("closed") logic.

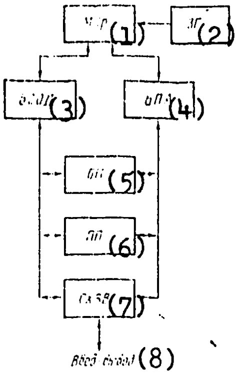


Figure 6-12. Generalized block diagram of a microcomputer.
Key:
1. Microprocessor MPr
2. Master clock ZG
3. Data exchange buffer
4. Address transmission buffer BSPA
5. Operational memory OP
6. Permanent memory PP
7. Information input-output circuit SkhVV
8. Input-output

In the field of control, microcomputers exert their greatest influence on changing the principles for designing local automatic units by expanding their capabilities substantially and, consequently, extending their areas of application. At the present time, the following prospective areas for the use of microcomputers can be delineated: 1) automated systems for monitoring and controlling technological processes and installations; 2) communication control systems; 3) multiprocessor computer systems; 4) information-measuring equipment; 5) control systems and units for domestic use; 6) controlling peripheral blocks for general-purpose and control computers.

Table A8-1 gives the specifications for several types of foreign microcomputers [65].

It should be mentioned that although control system algorithms are relatively small in volume, they must be realized on a real time scale. This requirement substantially complicates the use of computers with slow operating speeds in control systems for technological processes and objects. Therefore, we should also expect a further increase in the operating speed of UVM's.

UVM software has followed a path from the very simplest system for controlling a sequence of operations according to a program written in a machine-command language and realized in computers with a fixed structure all the way to extremely complex systems for organizing information processing in a multiprogram mode in modular systems of computer hardware.

As with general-purpose computers, problem-oriented languages together with a programming automation system are used extensively in UVM's. For all practical purposes there is not a single UVM that is not equipped with some kind of programming automation system.

Although in the 1960's such systems were based on machine-oriented languages, at the present time there is a tendency toward greater use of problem-oriented languages with differentiation of the areas of application.

At the same time, the use of external software based on programming computers or even programming in computer commands is typical of small UVM's. The latter is implemented primarily for UVM's operating in control systems for technological processes that have relatively uncomplicated information-processing algorithms and great stability when problems are being solved.

In view of the fact that there does not exist a general-purpose language for programming control problems, there is a tendency toward the development and use of special languages that meet the requirements for the solution of specific control problems in the best possible fashion. This tendency will apparently also be preserved in the future. At the same time, we should expect wider use of standard languages such as FORTRAN, ALGOL and COBOL, because as the areas of utilization of these languages expand, their penetration into systems for programming control assignments is inevitable. Naturally, for this purpose it is necessary to include in the standard languages the series of operators needed to enter the operations that are specific for control assignments.

There is a well-known joke: "software tries to load all the accessible memory and time of a computer." In this joke, which was formulated as "Parkinson's Third Law," there is a very large grain of truth. Therefore, both now and in the future we should expect to see it manifested to a greater degree, and we are already seeing a tendency toward the realization of an operating system's assignments by means of adding equipment. Such a solution is particularly essential for UVM's operating on a real time scale, because it makes it possible not only to reduce the volume of the memory, but also to increase computer productivity substantially by eliminating from the computation cycle the use of the slow peripheral units on which an operating system's programs are usually realized.

The areas of use of UVM's are extremely broad. The area in which they are most used is organizational and administrative control systems. On the average, it can be assumed that of 100 percent of the UVM's used in control systems, 70-80 percent are used in different types of automated organizational and administrative control systems and only 20-30 percent are used in systems for controlling technological processes. This trend will apparently be maintained in the near future. The basic reason for such a ratio is the relatively low reliability of modern UVM's.

Since a control system for technological processes and objects is realized in the form of automatic systems, the reliability of these systems' control units must be extremely high. It is also quite difficult to realize the requirement of operating on a real time scale. In automated control systems, these requirements are less stringent. Besides this, the greatest effect is achieved by the use of general-industrial-type UVM's in an automated organizational and administrative control system, since the problems solved by them require the processing of large volumes of information according to comparatively complex algorithms, the realization of which is impossible without the use of computers. The use of small UVM's to control technological processes should be expected in ever greater volume.

With the appearance of highly reliable mini-UVM's and microcomputers, the area of utilization of control computers will shift more and more into the area of automatic systems and local automatic installations.

In conclusion, we should point out yet another area of UVM use. Using UVM's as a base, it is comparatively simple to realize digital-analog complexes (TsAK). Such complexes contain analog and digital (control) computers. TsAK's are used most effectively to work out algorithms for digital control systems. In connection with this, as a rule the analog computer is used to model the controlled object (or process), while the digital unit realizes the control algorithms. Such TsAK's are realized relatively simply on the basis of UVM's, since the latter have devices for communicating with the object as part of their makeup.

Similar TsAK's make it possible to carry out laboratory processing of algorithms of the most complicated control systems with

UVM's and this, in the end, leads to a substantial shortening of the amount of time needed to introduce control systems as a whole.

11746
CSO: 8144/1911 - END -